DESIGN AND DEVELOPMENT OF A DATABASE SYSTEM FOR MANUFACTURING PLANNING AND CONTROL IN A MAKE-TO-STOCK ENVIRONMENT

by

MATHEW CHERIAN

DEPARTMENT OF INDUSTRIAL AND MANAGEMENT ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1993

DESIGN AND DEVELOPMENT OF A DATABASE SYSTEM FOR MANUFACTURING PLANNING AND CONTROL IN A MAKE-TO-STOCK ENVIRONMENT

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

MATHEW CHERIAN

to the

DEPARTMENT OF INDUSTRIAL & MANAGEMENT ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1993

C E R T I F I C A T E

30/4/93
K.N.

It is certified that the work contained in the thesis entitled **DESIGN AND DEVELOPMENT OF A DATABASE SYSTEM FOR MANUFACTURING PLANNING AND CONTROL IN A MAKE TO STOCK ENVIRONMENT** by MATHEW CHERIAN has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Kripa Shanker)
professor
Industrial and Management Engg.
Indian Institute of Technology
Kanpur, 208 016.

April, 1993.

ABSTRACT

Manufacturing planning and control (MPC) deals with all activities from acquisition of raw materials to delivery of completed products. Naturally, it involves a number of elements which generate and use data. Hence database, as repository for stored data which is integrated and shared by different users, forms an important ingredient of the information system for various decision making problems in MPC.

The present work looks into the issues with common database vis-a-vis integration of several stand alone databases for MPC. While common database for MPC are overwhelming complex, lack of standardization poses problem for integration for several stand alone databases. As a solution to these issues, decomposition of conceptual schema in align with various organizational functions is put forth by bringing in the concept of ownership of data. For this, "output-based" organizational decomposition is suggested.

Based on this, conceptual schema is developed for state-independent and state-dependant part of the database required for MPC in make-to-stock manufacturing environment and forms the basis for the implementation of database for MPC. Database is developed in relational database management system (RDBMS), INGRES version 6.2/03 hp installed in the main frame HP 9000/800 in IIT kanpur.

ACKNOWLEDGEMENTS

First and foremost I express my heartfelt gratitude and sincere thanks to my thesis supervisor, Dr.Kripa Shanker for his expert guidance, valuable criticisms and helpfulness throughout these days.

I sincerely thank Mr.Prasad and Mr.Nirmal Roberts of computer science dept. for their help to me in learning and mastering INGRES. Mr.Joshi and Miss.Anjali of TCS and the staff of DP cell for extending their cooperation in my effort to get acquaintance with INGRES, especially during in the initial stages.

I miss words to thank Mr.Bimal Modi for his concern and help rendered to me especially in the eleventh hour.

I have met some of the most interesting persons here in IME. The excellent cooperation of the staff of IME dept. helped me a lot in making my life here easier. Rajan, Saibal, Prabhakar, Rajesh, Rakesh, Sreekanth, Misra, Siva, Neeraj and kuttubuddin have really made IME affairs a memorable one.

I owe to Director of Technical education and former principal of institute where I am working, prof.Lebba for sponsoring me to this programme.

Lastly but not least, my thanks goes to my wife for the constant encouragement during these days when I was away from home.

April, 1993 M.C

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 MANUFACTURING PLANNING AND CONTROL (MPC)

Manufacturing planning and control system deals with all of the activities from acquisition of raw materials to delivery of completed products. MPC systems are also designed to support key management interfaces and activities in various functions of the firm such as cost accounting, order entry and customer service, logistics, budgeting, capital budgeting and strategic planning. Successful implementation of MPC system requires planning the correct timing and quantities of purchased and manufactured parts. It also involves the detailed execution of the plans and controlling the activities to conform to the plans. A good MPC system should be, amongst several advantages, able to provide the firm better responsiveness to customer requests, reduced times for a product introductions, improved quality levels, better strategic focus, etc.

The earlier attempts for a systematic approach for manufacturing planning and control (MPC) were made by engineering scientists like Taylor, Gnatt, Harris, Gigli, etc. The models were mostly based on the practical situations encountered at that time. The resulting models were rather simple, touching only fewer aspects of MPC. However, the insights contained in these simple models provided valuable contributions to solving production control problems in those days.

Later, especially during 50's and 60's, the developments in the field of MPC were along two lines. The first line concentrated on solving models and hardly dealt with the problems at the factory level. The works of Clark and Scarf (1960) on the structure of optimal ordering policies in multi-echelon stock-points, of Wagner and Whitin (1958) on the optimal

batch size determination in situations with fluctuating demand are some of such examples. There are hundreds of such papers reported in literature. The second line of research was based on the actual problems encountered in practice at the factory level. This resulted in the classification of the entire field into a number of problem oriented models and techniques. Works of Brown (1967, 1977), Burbidge (1971), VanHees and Monhemius (1972), Magee (1958) and Melese (1967) are some of the literature in this field.

Both of these lines did not lead to any comprehensive and complete scheme of classification, models and technique for MPC. The former withered under practical problems as it didn't have enough roots in practice. The latter was confused too much by the overwhelming complexity of the production problems, since they didn't have theoretical background. Bertrand, Wortmann and Wijngaard [1] provide further details on the historical development of MPC.

A natural consequence of these has been the recent development of approaches integrating the two lines of work. Some of the major approaches or principles that developed in 70's and 80's are: MRP (Material Requirement Planning), JIT (Just-In-Time) and OPT (Optimized Production Technology).

1.2 SOME RECENT APPROACHES

(1). Material Resource Planning (MRP)

The MRP essentially is concerned with

(a) determination of what final products, or the equivalent of final products should be produced at what time (later called the master production schedule)

(b) calculation of the required quantities of subassemblies, components and material, based on an up-to-date bill of material (BOM), the available inventories and the work-in-process, the batch size and the

manufacturing and purchasing lead times.

The MRP logic is detailed in section 4.3.6

(2). Just-in-Time **(JIT)**

The basis of JIT is to have throughput time short enough to meet any demand of finished goods and to eliminate the redundant stock or work-in-process as far as possible. The implementation of such system requires drastic reduction in set-up times and yield variations, small batch size, improved quality control methods, high commonalty of components for various final products and employee participation. The result of these is a system which is flexible and operates to produce only the necessary products at the necessary time in the necessary quantity with minimal flow times. There is a drastic reduction in the information processing and requirement (what is called removal of the 'hidden factory' [24] associated with MRP system) in JIT methods

(3). Optimized Production Technology (OPT)

The essence of OPT approach is to consider the efficient use of scarce capacity resource with clever scheduling. The objective is to reduce throughput time (T), inventory (I) and operating expense (OE). It starts with "forward" scheduling of bottleneck capacity in an efficient way considering market demand and its own potential. This schedule is the master for the "demand" placed on other capacities and its scheduling (called backward scheduling).

Vollmann, Berry and Whybark [24] have dealt excellently all three approaches and their subsequent developments. Romeo and Esparrgo [21], Rao and Sahergy [25] and Yamoshina [28] also deals with JIT. Cohen [5] and Goldratt [12] deals OPT approach.

Each of these approaches has its benefits and is suited for a particular environment. For instance, with MRP approach, it is difficult to model a situation where there exist serious capacity bottlenecks, which

must be scheduled at a high level in the control hierarchy. At the same time uncertainty plays an important role in the control process. The extension of MRP to MRP-II by means of the introduction of the Master Production Scheduling concept has provided some improvement [1].

In a repetitive manufacturing system, a well engineered JIT has an upperhand over MRP approach. In non-repetitive manufacturing environment, the application of pure JIT is difficult (Harhen [12]).

If a manufacturing planning and control system is solely based on the principle that efficient use of bottleneck capacity is crucial, then it is difficult to give production-order throughput time control and its use in long lead time item procurement (which is essential in the MRP) its due place and weightage [1].

For each particular production situation, the various relevant aspects of MPC must be weighted, balanced and integrated. However there has to be a general frame work (both from organizational and conceptual point of view) which does not deny the unique character of each particular situation, but which helps to identify it.

1.3 TYPES OF MANUFACTURING ENVIRONMENT

Manufacturing environment can be differentiated based on those that produce goods for inventory (make-to-stock) and those that produce based on customer order (customer-order driven production).

The make-to-stock organization produces in batches, carrying finished good inventories for most of its end items, and often produces consumer products and supply items for industries.

The customer-order driven manufacturing environment can be a

(i) assemble-to-order: those that assemble a wide variety of finished goods from a smaller set of standardized options from which customer may choose. It is similar to make-to-stock, but final assembly is based on customer order. Hence it is often classified under make-to-stock

manufacturing.

(ii) make-to-order: those that manufacture each customer order on a unique basis, without any standardized products from which customer may choose.

(iii) engineer-to-order: this is similar to make-to-order, but engineering design and the development of bill-of-material for each order are carried out by the organization.

In practice, there is nothing like pure customer-order driven organization. All types of organization do produce and stock some standardized products [2].

1.4 SCOPE OF PRESENT WORK AND ORGANIZATION OF THESIS

The objective of the present work is to design and develop database for manufacturing planning and control (MPC) in a make-to-stock manufacturing environment. The emphasis has been given to the development to the organization model and conceptual schema (conceptual data model) for a medium size discrete manufacturing firm, consisting of number of plants producing various end products. However, it may be extended to other types of firm.

Chapter 2 discusses the various issues connected with database for MPC and suggests organizational and conceptual decomposition as a solution to these.

Chapter 3 deals with the product types and the conceptual schema for state-independent part of database.

Chapter 4 describes the three levels of manufacturing planning and control and their conceptual schema.

chapter 5 looks into to the implementation aspects.

Conclusions are drawn in the last chapter.

CHAPTER 2

SYSTEM ORGANIZATION AND DESIGN CONSIDERATION

2.1 INTRODUCTION

Like other corporate resources (such as money, material, equipment and people), information is a critical resource and must be managed and controlled for effective decision making. Databases as repository for stored data which is integrated and shared by different users (Date [8]), forms an important ingredient of information system. This chapter deals with the organizational and conceptual aspects of design requirements for the database of manufacturing planning and control (MPC) system.

In section 2.2 of this chapter the issues of having a common database vis-a-vis integrating several stand alone databases is discussed. While the common database is becoming increasingly complex, lack of standardization poses problem to the integration of stand alone databases.

Structural decomposition and flexible integration of the organization as well as the conceptual data model are proposed in section 2.3, 2.4 and 2.5. The former emphasizes the need for modularizing into non-overlapping units, with well-defined tasks, responsibilities, data and frame of references. The latter stresses the need for interfacing these modules with well-defined co-ordination procedures, common data and consistent overlap in frames of reference. The decomposition at the organizational and conceptual levels is considered in sections 2.4 and 2.5 respectively.

2.2 DATABASE FOR MPC AND RELATED ISSUES

2.2.1 Common database

In a common database, the storage of data is centralized. Some of the advantages of centralized control of the data as in a common database are following: redundancy can be reduced, inconsistency can be avoided, the data can be shared, standards can be enforced, security restrictions can be applied and integrity can be maintained [8].

Following are the issues with common database for MPC

(1). Complexity

In MPC, the problem with common database is that they become complex due to large number of number of entity types and their complex association. For instance, market trends force all kinds of manufacturers now a days to offer a greater variety in products. For materials management and production, however, the higher commonalty in products improves manufacturing efficiency. To bridge the gap, it is customary to derive many different versions from existing products. Thus most of the end-items have a number of features and for each feature, several options may have to be chosen (by customers). Due to the explosion of the number of different combinations of options, the number of end-items can be too large to define a manufacturing bill of material (BOM) for every single end-item. Also, if sales volume is low, forecasting of individual end-items may not be feasible, but some sort of grouping is required to facilitate group forecasting. For example, a final product characterized by five features, two of which have five options and three of which are extras and are optional can result in 200 (i.e. 5^2*2^3) different final products. In the real-life situation (especially in assemble-to-order manufacturing environment), there would be millions of such final products. This causes problems in forecasting requirements, BOM storage

and maintenance in the database for commercial planning and control.

(2). Heterogeneity of users

Heterogeneity of the users is another problem with common database in MPC. This brings difference in (a) "Views" on the data and (b) relevance of attributes of entity types and the relationship between them. For example, "manufacturing people" and "the material planning people" will have different view on bill of material. Whereas the former needs detailed product structure which includes transient items ("phantom items", (Orlicky [17])), the latter plans the material requirement by "jumping over" these transient items.

(3). Difference in the aggregation and quality of data required

The difference in (a) aggregation levels required (e.g. as one go up in the management level, the data required is in more aggregate form) (b) the ways of aggregation of data (e.g. for a production plan, sales people needs aggregation based on quantity of final products produced by that plan, whereas manufacturing people looks for total labour hrs, amount of capacity required at the critical centers or types, and (c) requirement of actuality (e.g. shop floor people requires actual status of each production orders, capacity center, etc.) add to the complex demand of data expected from the common database.

(4). Responsibility of the correctness of data

As more and more organization functions are sharing data, it becomes nearly impossible to assign responsibilities for the correctness of the data in common database. In other words, the question is who is authorized to make what changes in data and what are its consequences for the applications (can be a computer program or some manual procedure) using these data. The problem, for example, when engineering changes require a great number of approval meetings, where nearly all functions

like product definition (PD), process planning, shop floor control, etc are involved, the decision procedures for affecting changes in database take a lot of time and coordinating effort. Even after the approval, engineering changes, nevertheless, may remain cause problems because the meaning of all attributes in the database and its consequences for applications may be far from clear.

(5). Difficulty in development & maintenance of application software

Each new application assumes a certain semantical structure of the data (the semantical aspect refers to the meaning of the symbols and sentences in which the information is expressed). Therefore, the meaning of the data is often specified by the existing application software and a change of this meaning is hardly possible. This results in difficulty in the development of new application programs. For example, often in many firms, the bill of material is available in the form of detailed assembly/manufacture structure, describing how the items are to be manufactured using the facilities at various production divisions. This detailed information is often not suitable for material coordination, which requires an aggregated BOM, jumping over the transient or phantom items. If the necessity for such an aggregation has not been specified beforehand and an aggregated BOM could not be derived either, then it posses further difficulty in the software development for material coordination later.

Secondly, in a common database, a data element can be modified by various application programs. Hence it is very difficult to understand the cause of errors, if any, and to control the effects of modifications.

These issue with common database of MPC resulted in, what Pels and Wortmann [19] have described "data administration crisis".

2.2.2 Issues with integration of stand alone databases of MPC

With lowering of cost of computing power and data storage, many local databases are introduced as a part of stand alone applications for various end-users. They are usually embedded in local computers. In the context of MPC, local databases are introduced for application functions such as order processing, master scheduling, inventory control, MRP, computerized scheduling, etc.

Data communication facilities are mainly concerned with the technical aspect of integration of databases and related information systems, i.e. registering and transporting data and messages at different locations without errors and without delay. Standards like OSI and MAP aim at easing integration at the technical level.

Standardization and advances in distributed database management systems (DBMS) offer the applications of different subsystems to access to each others data, regardless of the physical location of the data the different storage and access methods used and the differences in data manipulation languages (DML) of the local DBMS.

However in production system, as mentioned earlier, these applications have different frame of references and views. Integration at the semantical level is concerned with the problem that sender and receiver must attach the same meaning to the message they exchange or the same kind of the data registered in their databases. Misinterpretations cause uncoordinated behaviour of the cooperating functions. This happens when, for instance, an angle is expressed in degrees, but interpreted as radians by the receiver function. Another problem may be that a message or data which is meaningful to the sender may be nonsense to the receiver. An example of such a communication clash occurs when sender refers to angle of, say 400 degrees, while receiver expects only value between 0 and 360.

The result is that the system must be stopped and reset. Yet another problem in semantical integration is that the meaning of a message does not depend upon its contents, but may also depend upon the actual state of the receiver. A typical example would be that for machine loading, one needs to know the actual state of machine, i.e. what all processes are currently being done or are in the queue on that machine. In short, consistent integration at the semantical level requires not only precise definition of the contents and meanings of all possible messages or range of data values, but also that the communicating or integrating components should have sufficient knowledge of each others actual state [18].

2.3 Decomposition : A Suggested Approach

Distributing programs and data as well as creation of common database will not resolve the "data administration" crisis. Interconnecting distributed computers will not result in an integrated system. Structured decomposition and flexible integration is proposed as a solution to these issues. It involves the decomposition of the total organizational function into sub functions. Each such function of the system has distinct and well defined tasks, responsibilities, data and frames of reference. The flexible integration means that these functions are interfaced by well-defined coordination procedures, common data and organized control over areas where it requires overlap in frames of reference [19].

2.4 DECOMPOSITION AT THE ORGANIZATIONAL LEVEL

As per contingency approach (Galbraith, [10]), the central question in organizational design deals with the ways in which complexity and uncertainty in executing tasks are reduced. When organizations face an

insufficient performance (such as poor data-management or slow software

Alternative / Effect	Creation of slack resources	Creation of self contained task	Investments in vertical information system	Creation of lateral relationship
Reduce the need for information processing	x	x		
Increase capacity to process information			x	x
Change organizational structure		x		x
Decrease required level of performance	x			

x : indicates the applicable effect of the alternative

Fig 2.1 Design strategies for reducing task uncertainty and complexity

development), it can choose one of the several alternative strategies as shown in the Fig 2.1. ([10], [19]). This section discusses various alternatives and ends with a description of organization structure.

(a) Slack resources

The first alternative, i.e. creation of slack resources, is usually what happens if there is no active response to organizational problems. In many organizations, this takes the form of time-slack (e.g. the decision procedures take a lot of time) or inventory buildup due to lower information processing. This will, naturally, decrease the performance level. This happens mainly in engineering change problem described earlier.

(b) Self-contained task

The creation of independent unit or what Galbraith [10] has described "self-contained task", that can perform all the necessary aspects of a functional task internally will reduce the need for coordination (i.e. the need for information processing is reduced). However if the organizational structure is based primarily on clustering of similar skills, resources or professions (Pels and Wortmann [19] has defined as "input-based task design"), it would necessitate quite some effort to coordinate ongoing process that share resources from several functional task areas. A typical example for this is an organization structure based on process layout setup. On contrary, in an "ouput-based" organization structure, the grouping should be based on different skills and resources to fulfill a certain mission such as production of a certain type of product. In the production area, self-contained tasks may take the form of group technology at the shop floor level; product divisions at corporate levels, etc. Such discretisation of organizational structure follows a decentralization of staff skills and a distribution of resources. However in output-based organization, the problem emerges of how to efficiently utilize labour and other resources. This calls for increase information processing capacity and creation of lateral relation between these units.

(c) To increase the information processing capacity

The third and fourth alternatives as shown in the Fig 2.1 are more or less complementary to each other as both increases the capacity to process information. The purpose to invest in vertical information system should aim at generating aggregated information for aggregated planning and control over the output of self-contained task units. This points toward an hierarchical setup as detailed later.

The creation of self-contained task unit brings discretion (or decision power) at lower levels of the organization, without the need of information from other groups, for solving its problems. However, if discretion is to be increased at lower levels without reducing resource sharing, lateral relations are required (Galbraith [10]). For instance, in material coordination in the production area, where one production division or unit delivers the resources (i.e. input) for another one, some form of lateral coordination is required; otherwise it may end up in the inventory buildup to reduce the uncertainty of the input for the latter unit. Lateral relations may take various forms; by direct contact between the managers involved (weaker form), or in the creation of task forces or permanent teams or appointment of project leader. If the coordinating aspect is critical, dual authority relations and a matrix setup may be established.

(d) Organizational decomposition and databases

"The most appropriate way to eliminate coordinating problems in decision-making and, therefore, in data management is to create self-contained tasks" (Pels & Wortmann [19]). This helps in the design of local databases and local applications for which the self-contained task group is responsible. For the remaining coordinating problems (between the various self-contained task group), a lateral relation should be created. If this lateral relation takes the form of a separate organizational entity, the responsibility for a part of the common database can be allocated to this entity. For instance, the material planning and control acts as "lateral relation" to coordinate the output of the various production units. Such an organizational design allows to allocate the responsibility for large parts of the database and the correctness of data contained in the database to separate groups. If the responsibility for

parts of the database is allocated to distinct self-contained task groups, it becomes possible to design vertical information systems by formally deriving information from "local" databases. This calls for hierarchical approach to the organization.

2.4.1 Hierarchical approach

The decomposition of organization of MPC into different hierarchical levels helps in putting planning and control function in proper perspective and easy handling of the whole situation (Meal [15]). The hierarchical approach retains the conceptual simplicity of the decentralized approach. Decisions which must be made at the corporate levels are centralized i.e. those which can be made locally are delegated to the lower level (say plant level). While the decisions at the top constrain those made below in a cascading fashion, the lower levels send performance characteristics and operating results in rather aggregate manner. Because of this, the hierarchical approach does not require large centralized database and computing procedures. It also allows the development of "local" databases at lower level.

Fig.2.2 shows major planning and control areas of a business. Business planning is concerned with where the business is to be driven during the next five or ten years (or even more). It involves dialogues and broad trade-offs between various management functions such as marketing, finance, production,etc. at the top level to fix the overall objective of the firm.

Master planning deals with planning the production to support the business plan for the next two or three years.

Material planning and control is the process of determining the materials (ordering and receipt) to support the master planning for the next six months to a year.

Production unit planning and control is concerned with how to get the product built with received material and schedule the jobs through various capacity centers.

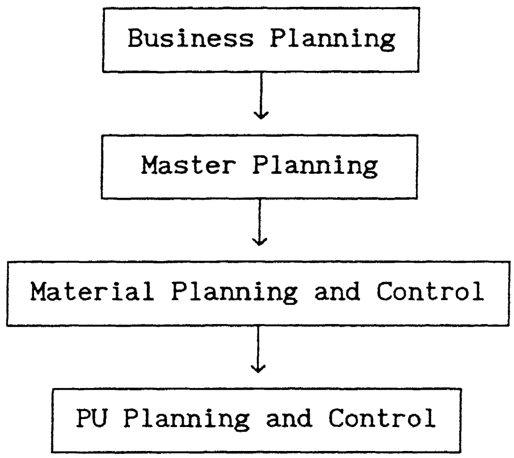


Fig 2.2 Hierarchy of planning and control in business

2.4.2. Structure of the organization

The elements of an organization and the relationship in their functions are discussed in the following paragraphs.

(a) Production unit (PU)

A production unit (PU) may defined as discrete unit which on short term is self-contained with respect to the use of its resources, which is responsible for production of a specific set of products (the production unit end-items) from a specific set of materials and components (the production unit start-items), may model itself and use technology which may or may not be same as other PUs. More precisely, a production unit is defined by a class PU-end items with for each PU-end item, a class of operations with corresponding material and resource requirements. A production unit may consist of large installation or machine or it may consist of an entire production or assembly hall with different types of machinery and personal. Several PUs may have the same type of capacity.

Capacity types can be grouped in different capacity centers, either on the basis of same machine type or on operation basis or any other consideration such as group technology [1].

The allocation of specific capacity rather permanently to PU for the production of specific PU end-items decreases freedom in the system. However, the introduction of PUs must offset the above deficiency by

(i) reducing the complexity of the decision problems

(ii) increasing the stability

(iii) creating improved models (both mental and formal) of the decision problems within the PUs and at other levels.

These must improve quality of organization in each of PUs as well as logistic decisions [1].

(2). Item orders and shop floor control

Orders for the production of items (item orders) are assigned to the PU for manufacturing batches of PU end-items. These orders are assigned after considering the limited availability of capacities and throughput time (time required for the completion of a production order) requirements. Once an item order is assigned to the PU, the various shop operations are scheduled to various capacity centers and is sequenced by some priority rules.

(3). Logistics

Logistics coordinates the material flow between production units and at balancing the capacity load of the production units. It also coordinate production and sales. In addition, this function is responsible for the supply of long lead time components. On the short term, logistic planning and control performs this function by usage of the product structure, while "jumping over" the phantoms (transient items). On the medium terms, it employs planning modulus (also known as product group

or product families) [9].

(5). Co-ordination of production and planning

One of the important functions of logistics is to co-ordinate production and sales. It involves customer order processing, determination of net demand of items (demand analysis planning) and the creation of master production plan This is detailed in chapter 4.

(6). Material planning and control

At logistics level, the control of PU end-items is done at aggregate level. and the production plan of the final products is generated. This is called Master Production Schedule (MPS).

From MPS, the detailed co-ordination (i.e. material co-ordination) of individual end-items of PU is done through well known MRP (Material Requirement Planning) logic which results in the creation of item orders at various time interval. For purchase items, the item orders initiate purchase order and related activities, whereas for manufacturing item, it is released to the concerned PU.

(6). Organization structure

The structure of organization will be hierarchical, consisting of self contained units called production units with lateral and vertical relations organized as logistics. The actual manufacturing takes place within a number of production units (PU). Each PU has number of capacity centers (CC) where machines or production equipment are grouped together.

2.5 CONCEPTUAL SCHEMA AND DECOMPOSITION

In the database language, the frame of reference or universe of discourse (i.e. the world about which data is stored) is modeled with what is called conceptual schema or conceptual data model ([7], [8]). It does by capturing the underlying real-world information structure through a set

of concrete or abstract objects called entities and their relationships. A class of similar entities is called entity class or type. Conceptual schema have been developed to support the design of common databases. Appendix A gives more details on various elements of conceptual schema and conceptual data diagram (used in this thesis). This section deals with the decomposition of conceptual schema.

The decomposition of conceptual schema means that it is to be split into non-overlapping modules, as much as possible corresponding to the self-contained units (like PUs) [19]. The module of a conceptual schema can be defined as a unit of update and query authority [18].

The first step to decompose a conceptual schema is to identify and define a set of modules. The second step is to assign to each module the subset of entity classes for which the module gives update authorization. This subset is called the own-domain of the module. The third step is to assign to each module the subset of entity classes for which module gives query authorization. This subset is called the view-domain of the module (i.e. visible entities) and it includes the own-domain also. The difference between view-domain and own-domain, containing the entity classes that may be queried only is called the foreign-domain of the module. Thus every entity class is owned by at least one module.

Ownership can be shared; two or more modules may own the same entity class. The own-domain of each module may be divided into two: the public-domain with the entity classes that are visible for one or more other modules. The data associated with such entity classes are called the public data. The entities in private-domain are visible for a module only (data associated are called private data).

The concept of own and foreign domain is explained here with help of the Fig 3.1 in Chapter 3 which shows the conceptual schema of BOM for

production unit planning and control. The Figure shows demarcation to highlight the owner function of each entities. For instance, entity phantom is in the own-domain of the product definition (PD) function. It can be public data if it is owned/visible by production unit management also (depending upon the policy adopted by the organization). However entities like prod-unit are foreign to PD function and are shown outside demarcation line shown for PD.

2.5.1 Ownership and responsibility of data

Along with decomposition of conceptual schema, the ownership of data is also to be introduced in order to link entity types with organizational functions. In a well designed organizations, it should be possible to point to one single organizational unit as the responsible task for the contents of any datum. For instance, in Fig 3.1, PD function is the owner of entities like phantom, manu-item, etc. They are placed inside the boundary line drawn for PD function. The situation where different functional task unit are allowed to modify the same data occurs mainly where these different units share a single resource. A typical example is bill of material. Even though it is usually under the responsibility of product definition function, any change has to be with the approval of other functional group such as process planning group, manufacturing group, etc. The modification of the structure of the public data requires some organizational coordination. This calls for a lateral relations.

2.5.2 Sub schemata

The sub schema of a module must show all the specifications from the global schema that are relevant for the user of the module. (Figures 3.1, 3.2, 3.3, 3.4, 3.5, 4.1, 4.5, 4.8, 4.9 and 4.10 are the data structure diagrams of various sub schema developed in this thesis.) It

must show all visible entities and their attributes of the module. Secondly, entities must further be classified into own and foreign domain with respect to the module; i.e. the modules that own the foreign entity classes and that "see" public entity classes of the own domain of the module are to be identified. Finally, it must show all constraints (see appendix A) that are relevant for the user. A constraint is relevant when it refers to one or more visible entity classes of the module.

2.5.3 Module independence

The knowledge of the entity classes in foreign domain and the relevant constraints help to check whether an update or deletion is in conflict with the actual state of the own domain of the surrounding modules because of some integrity constraint, causing communication clash [18]. For instance, in Fig 3.1, if an entity of the entity class item is deleted without "knowing" its relation with entity supp-item (an entity, that has referential relationship [see appendix A] with item) for that item, it produces inconsistent database state. Hence such a schemata must help in the development of application programs of the modules with the knowledge only of its sub schema (i.e. without the need of knowing the whole global schema) and without any communication clash when integrated with the global schema. A module with such a property is called an independent module.

Pels [18] has stated that sufficient, though not necessary condition for independence of a module in a modular conceptual schema is that all applicable constraints are visible for this module. A constraint is called "applicable" for a module if it refers to one or more own entity classes of the module. It is called "visible" if it refers to only entity classes that are visible for the module. The referential constraint between item and supp-item (entity for supplied item) in the previous

illustration is an applicable constraint. The knowledge of the referential constraint with the above two entities will help the application programmer to develop precautions before an item is deleted or updated.

Module independence is the basic reason for decomposition of conceptual schema into modules. The implication of the independence of module in an integrated system is that all applications that are designed with only knowledge of the sub schema of this module and tested upon the separated database module based on this sub schema will not produce any errors at the conceptual level when operated for the whole system [18]. This facilitates in the modular design and development of complex databases and related information systems.

2.5.4 Integration of modular schemata

In database design theory it is generally supposed that different schemata can only be integrated if they have equal specifications for equal elements [18]. For integration for conceptual schemata, it is required that equal entity classes have equal attribute sets. Schema that fulfill this requirement are called "compatible".

For the integration of modular schemata, it is also required that the specification of own and view-domain in both schemata must be compatible. It means that if module m and entity class e are common for both schemata, and e is visible/own for m in one schema, it must be visible/own for m in the other schema. Thus, in the integration of modular schemata, it is needed to identify entities that are in the own and foreign domains of module and retain the original modules as a recognizable unit.

Safe integration occurs when application programs that have been validated against the sub schema of the module, remain valid also for the integrated schema. For such a "transferability of applications" [18] under

integration, in addition to independence of module, it is also required that the constraints that each module imposes on its foreign data are no more severe than the constraints that are imposed by the owner of the data. For instance, as mentioned before, if a data element in the own domain of the module that accepts angular measurement up to 400 radians is imposed a restriction between 0 to 360 on integration is a constraint that is more severe than when the module is alone. Hence, when integrating modules, as long as no constraints are added that do not refer to any own entity class of that module i.e. applicable constraint, one can assume safe integration at the conceptual level.

2.5.5 Integration of stand alone databases

The concept of modular decomposition of conceptual schema can also be extended to the integration of various stand alone databases and related software driven information system. Pels [18] has suggested four steps for the same. Such an integration need not require time consuming integration tests since communication clashes can be predicted from an analysis of the conceptual schemata of the different components.

The first step is to make up the conceptual schema of each of the different components. By making this a modular schema it is possible to discriminate between the own data that is generated by this component and the foreign data that the component expects to receive from its environment.

The second step is to compare the structure of the foreign data of each component with the structure of the corresponding elements in the other components in order to locate possible inconsistencies.

The third step is to resolve the conflicts either modify the component or creating intermediary components or interfaces. Some of the possible conflicts are

(a) foreign data elements do not exist as own elements in the other components : a function can be implemented into the intermediary system to collect and supply these data, (b) the same data element is regarded as own by two or more components (redundancy) : one of the components can made the owner and others be modified to accept the data as a copy from outside, (c) for a common entity type the owner has more attributes or otherwise more detail than the receiver can accept : the intermediary system can provide a view upon the original data that provides only the projection, selection or aggregation that is required, (d) the receiver has stronger constraints than the owner : the additional constraint must be implemented as 'manual' constraints in the user procedures of the owner component, (e) a component appears to be not an independent module in the integrated system (i.e. applicable constraint that are not visible are present) : this requires modification of component by identifying such constraint.

The fourth step is to implement the communication procedures at the internal level. Distributed DBMSs and standard communication networks will reduce the difficulty of this task greatly.

2.6 CONCLUSIONS

In this chapter, the problems of common database and integrating several stand alone databases for MPC are discussed. Decomposition at organizational and conceptual levels is looked into as a solution for this. The organizational decomposition calls for the creation of output-based self-contained functional task units in an hierarchical setup, with organized unit for lateral relation for better resource sharing. The planning and control function is organized primarily into three hierarchical levels; production unit (PU) control (where actual

manufacturing takes place), material planning and control (where good flow between PUs are coordinated) and master planning (where sales and demand for the items is coordinated with production at an aggregated level). The conceptual decomposition aims at the modularisation of the conceptual schema in line with organizational self-contained task functions. This eases not only in fixing responsibility of data through the concept of ownership of data, but also in bringing flexibility in the design, development and maintenance of complex system through modular design of databases and information system.

CHAPTER3

CONCEPTUAL SCHEMA FOR STATE INDEPENDENT PART OF THE DATABASE

The "state" refers to the state of orders, resources, which are to be controlled. The state-independent part of the database constitutes all kinds of information relating to products, technology, manufacturing equipments and personnel, which is indirectly supportive to the recording and planning of orders and resources. It allows the capturing of data such as standard products, goes-into relation between products (bill of material), routings, capacity types, standard lead time and so on.

In the first part of this chapter conceptual schema for product types and bill of material is developed. The schema for process plan and facility informations are detailed later.

3.1 CONCEPTUAL SCHEMA FOR PRODUCT TYPES AND BILL OF MATERIAL (BOM)

Bill-of-material of an item may be defined as the list of components with the number of each of the component that "goes into" the item. Such relationship is called BOM relationship (or gozinto relationship). Essentially, such relationship consists of a "parent" item, a "child" item that goes into the parent and number of child item required. BOM is one of the most important information of a manufactured item. In a make-to-stock environment, BOM information, along with routing information, forms the basis of all manufacturing activities. Usually the development and maintenance of BOM and various items is under the responsibility of Product Definition (PD) function.

3.1.1 Conceptual schema of BOM for PU planning and control

1. Product types

The Fig 3.1 shows major product types with their owner function and their relationships. At the most abstract level, the entity type 'item' is identified. The attributes of this type are at the same time attributes for all other types that are considered specialization of the super-type item. An item can be either a manufactured item or a purchased item.

A manufactured item (manu-item) is manufactured in a production unit (PU), but PU can manufacture one or more manu-item. Process Planning.

A manufactured item can be either a manufactured product (called as product) or a manufactured phantom (called as phantom (orlicky [17])).

A product is defined in terms of (i) a production unit end item, (ii) manufactured stock items (i.e. these items are delivered to a logistic stock point).

A production unit manufacture products using other products and purchased items as inputs, which in turn may be used for further high-level product's manufacture.

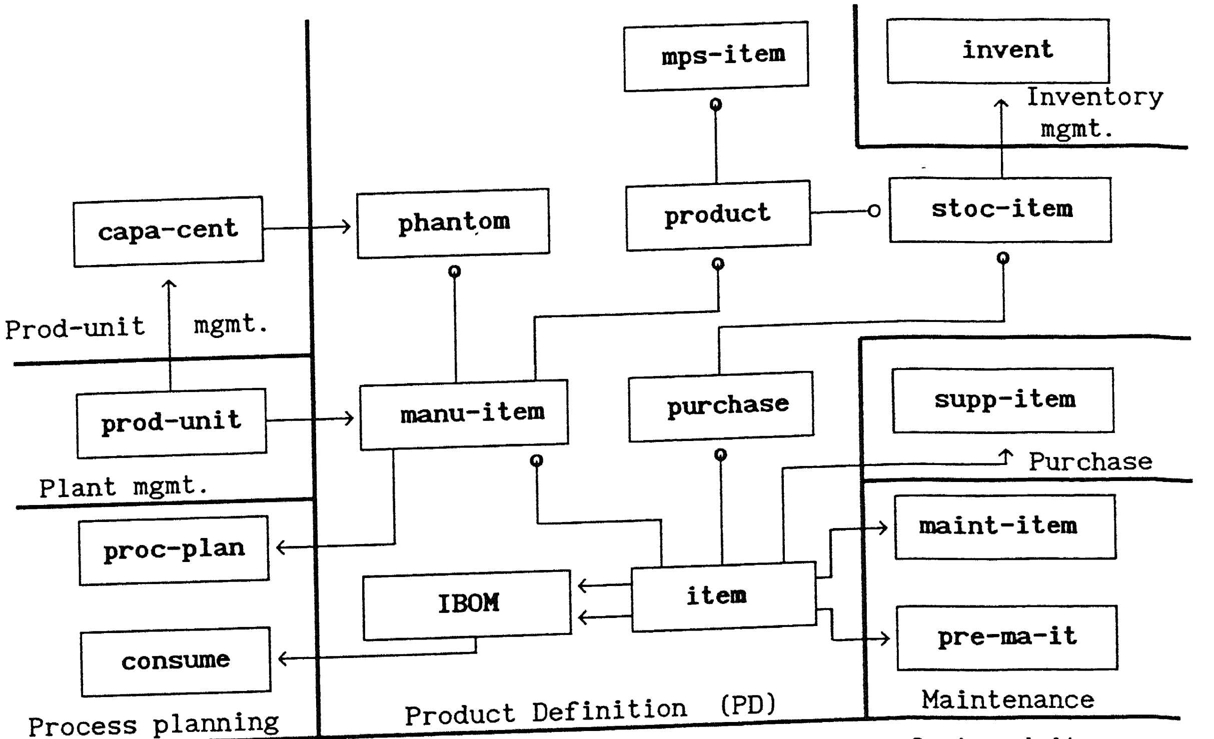
Within a production unit, a more detailed description of the products' manufacturing structure is used. These are transient items which may exist at some point in time during the manufacturing process, but which are not delivered to the stock point. Each phantom is identified by the capacity center, which produces it, inside a production unit.

Now, viewing from a different angle, all products and purchased items may be termed as logistic stock item or simply stock item. A stock item has certain distinctive attributes, like stock locations, quantities, safety stock and its inventory status and storage (identified as entity invent) are under inventory management. Thus a stock item can be either (i) product (i.e. manufactured Product) or (2) purchased item.

2. Relations

For every manufactured item, there must be at least a 'child' (or goes into) item(s). This is traditionally known as bill of material relation. Here it is termed as Item Bill of Material (ibom).

Each item can be supplied by more than one supplier and this relation is depicted in the entity supp-item.



prod-unit - production unit

capa-cent - capacity centre

proc-plan - process plan

item - physical item

invent - inventory status and storage

IBOM - item bill of material (item_bom)

supp-item - supplier and item relationship

maint-item - items used for maintenance

manu-item - manufactured item

purchase - purchase item

product - manufacured product

phantom - phantom item

stoc-item - stock item

mps-item - master production scheduling item

consume - required item for a normative operation

pre-ma-it - items for preventive maintenance

Fig 3.1 Conceptual schema of BOM for production unit planning and control

3.1.2 Conceptual schema of BOM for logistic planning & control

1. Product types

Fig 3.2 shows the conceptual schema of BOM for logistic planning and control. A planning module can be defined as the family of physical items that consume approximately the same quantity of the same type of resources. A "resource" can be anything, like cost, labour hour, floor space. It can be classified as

(i) Product family (Capacity planning module)

(ii) Purchase family (Material planning module)

A product family (also called by different authors as 'production family' or 'product group') is used for Master Production Planning (MPP) and is formed by grouping Master Products for scheduling. The logic for grouping varies based on the needs of individual companies. It may be structured based on

(a) Exactly the same as a sales family (based on sales trend or customers.....)

(b) Based on cost groups (high-cost items, medium cost items, low-cost items and so on)

(c) Based on a grouping of the consumption of similar machine and/or labour hours

(d) Based on a distinction between standard and customized products (i.e. standard final product and customer specified final product)

(e) Based on some unique material or component requirement etc.

As in the case of product family (capacity planning modules), the logic of classification of purchase items into various purchase families (material planning module) varies with the need of the individual companies. It may be structured based on

a) cost groups (i.e. high, medium, low,...)

b) vendors who supply these items

c) lead time (high, medium,low,...)

A Master Product for scheduling (MPS item in MRP-II literature) are manufactured products at the highest level in BOM and the planning levels of other manufactured items can be derived from it with help of BOM and MRP logic.

2. Relations

The various relations in logistic control function module are

(a) Material planning module make up (mapm-make).

A purchase item unlike product can belong to many different material planning modules. This relationship shows what purchase items consists of a material planning modules and their mix percentage.

(b) Commercial Family-Planning Modules Makeup (cfpm-make).

The commercial division has to communicate its forecasts and plans to the planning division (i.e. logistic control). In order to do so, the relationship between the commercial families and planning modules is required; which is of N:M relationship. It can be considered as a matrix A(i,j) indicating the expectation of the production volume of a specific planning module j, when the sales volume of a family i is given.

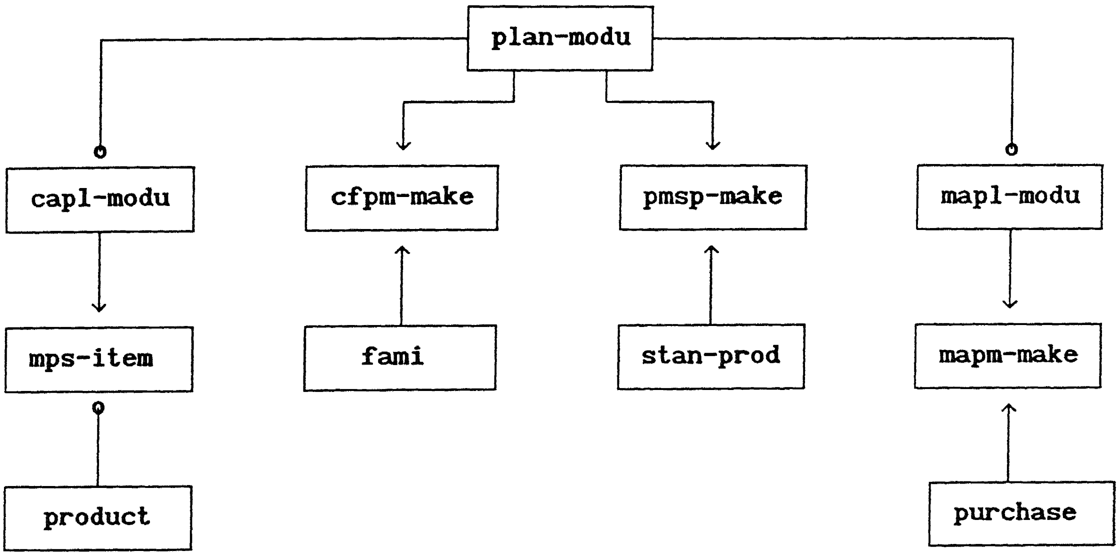
(c) Planning Module-Final Product Makeup (pmsp-make)

A similar consideration, holds for the relationship between final products and planning modules. This relationship is also a N:M relationship and is used by the logistic control function to compare the actual demand for final products with the forecasted demand for planning modules. To standard final products, pmsp-make holds the information and is of gozinto type relationship. For customer specified final product, each actual demand is broken down to demand of its components and is defined through customer specified final product makeup relationship (entity is cusp-make). This data can be aggregated at product group level

to get product level actual demand.

(d) Mix percentage of MPS-items within a product groups

The forecast at commercial family level is brought at planning module (product group) level. It is further exploded to MPS-items in order to have realistic and reliable projected demand quantities. The mix percentages are stored in the entity mps_item.



plan-modu - planning module (product family)

capl-modu - capacity planning module (product family for product)

mapl-modu - material planning module (product family for purchase)

cfpm-make - commercial family-planning module makeup relationship

pmsp-make - planning module-standard final product makeup relationship

stan-prod - standard final product

product - manufactured product

fami - commercial family

purchase- purchase item

mapm-make - material planning module-purchase item makeup relationship

mps-item - master product for scheduling

Fig 3.2 Conceptual schema of BOM for logistics planning and control

3.1.3 Conceptual schema of BOM for commercial planning & control

1. Product types

A commercial item can be either (a) commercial family or (b) commercial module.

A commercial family is defined as the family of final products. In other words, every final product should precisely belong to one commercial family.

A commercial family consists of two or more (first level) alternatives, called commercial modules. A commercial module consists of a number of manufactured products and purchased items, together with a number of (second level) commercial modules. A commercial module may be called as generic item, since it describes a range of items. A generic item can be of two types.

(a) Direct: A generic item represents a limited set of specific items. It is made specific by replacing it with one of the specific items.

(b) Indirect: A generic item may be generic just because one of its components at any lower level of its BOM happens to be generic (either directly or indirectly). It becomes specific indirectly, when all of its components are specific [23].

A Generic Bill of Material (Generic BOM) describes a range (genus) of products (i.e. in a commercial family) as against a specific Bill of material (specific BOM) which describes exactly the product. Hence each commercial family has a generic BOM and is not directly to be used for planning or manufacturing purposes. Rather it is only a frame work for creating a specific BOM at the time one is needed.

If any one of the item in BOM is generic, then that BOM is generic. Hence, a generic BOM, like generic item, becomes specific (rather a specific BOM is derived from generic BOM) when all its generic items are

As mentioned earlier, a commercial item can have zero or more specific items. A specific item has to be either a manufactured product (mostly it is so) or a purchased item. A final product(s) is the output of a manufacturing organization as a whole. It can be

(a) a standard final product, which invariably has to be a stock item. Usually it is a product, but it can be a purchase item also (for e.g. spare parts, manufactured outside the organization)

(b) customer-specified final product is based on special requirement of the customer and the order that is placed.

The attributes of each item-type are different.

2. Parameter and parameter values

A set of parameters P_i $\{P_1, P_2, \ldots . P_n\}$ representing features of a commercial family is defined under parameter. Now each parameter P_i of this set may have a number of values representing the options. The values for parameter P_i constitute a set $V(P_i : \{PV_{i,1}, PV_{i,2}, \ldots ., PV_{i,q}\}$. The set of parameters and their values must be defined in such a way that, if all parameters of a commercial family are given a value, a set of parameter values $C:\{PV_{1,s}, PV_{2,m}, \ldots \ldots PV_{n,t}\}$ is obtained uniquely identifying a final product within a commercial family. Now this set of parameter values in combination with the generic BOM implicitly determines a specific BOM representing the defined product. The Figure 3.3 shows the conceptual schema of the tool which transforms this implicitly defined BOM into an explicit one [23].

3. Condition

The substitution of generic items by specific ones is controlled by the identifying set of parameter values. For each generic item it should be decided whether, or which of, the specific items must be selected and inserted into the specific BOM. To have the selection of specific items be controlled by the set of parameter values, each of these items has to be

related to a boolean expression. This expression must be defined in terms of parameter values. Depending on these parameter values, the condition is either FALSE meaning the specific item is to inserted if that parameter value is not chosen in the set C (defined earlier) or TRUE meaning the specific item is to be inserted if the parameter value is present in the set C.

The boolean conditions related to BOM relationships are more complex. A specific item can relate to more than one parameter values. The relation between a specific parameter and different parameter values can either by conjunctions (i.e. AND) or by disjunctions (i.e. OR).

So a specific item is selected if and only if following conditions are satisfied.

(a) All condition which relate AND related parameter values to the specific item are satisfied.

(b) At least one of the condition which relate OR related parameter to the specific item are satisfied.

(c) A condition is satisfied if the true-value

(i) TRUE, then the parameter value related to that condition must be present (or chosen) in the set of parameter values identified for the chosen commercial family.

(ii) FALSE, then the parameter value related to that condition must be absent (or not chosen) in the set of parameter values identified for the chosen commercial family ([1], [23]).

The example given in the Appendix B illustrates the 'CONDITION'.

4. Constraint

Some parameter values could be mutually exclusive. It could be

(a) inter parameter value mutual exclusiveness.

Once a parameter value of a parameter is chosen, the rest of parameter values of the same parameter stands unqualified for further

choosing. For e.g., in the examples given in the appendix B, choice of red colour for holder colour (parameter) excludes the parameter values blue and green (i.e. parameter values).

(b) intra parameter value mutual exclusiveness

Suppose it is not possible to have red holder with round top for some reason (it could be technological constraint, or financial constraint, or due to same other reasons). So once red holder is chosen, the parameter value round of the parameter top shape becomes a non valid parameter value.

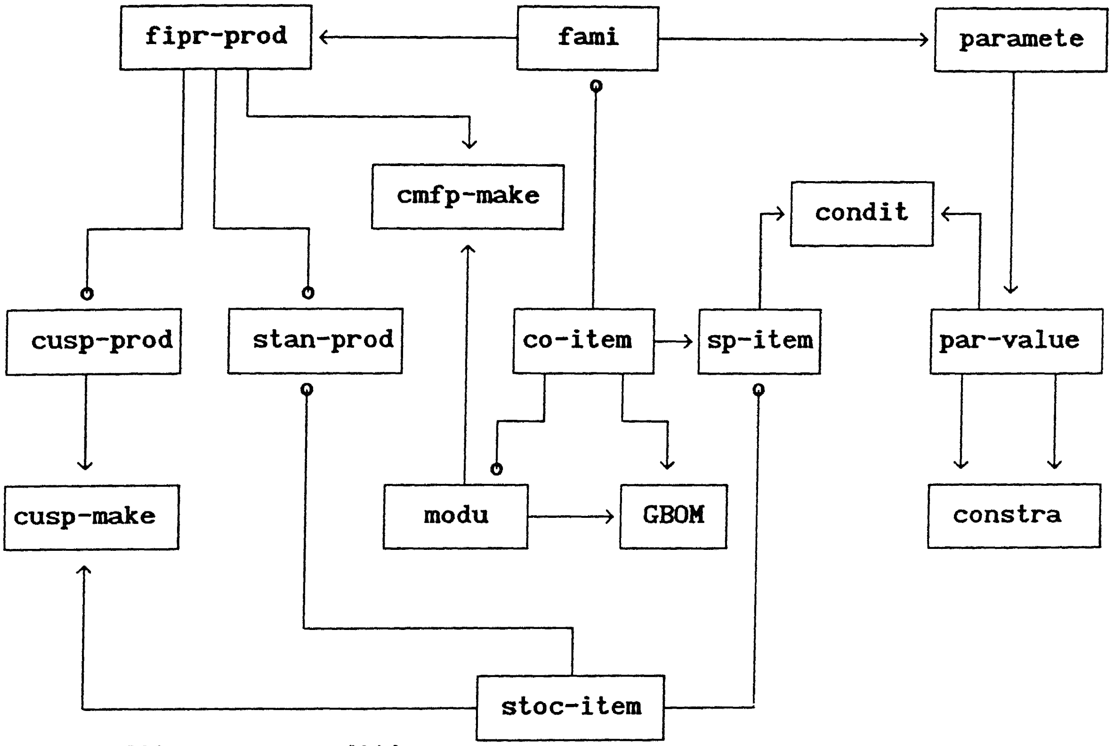
These constraints are between parameter values. The truth value false in a constraint indicates that the S-parameter (subject parameter) value is mutually exclusive to O-parameter (object parameter). The truth value True indicates that the choice for the one parameter value (S-parameter value) is valid only if the other parameter value (O-parameter value) is chosen. In other words the subject-parameter value is open for choice subjected to the constraint that object-parameter value is chosen. It does not mean that subject-parameter be always chosen ([1], [23]).

5. Relations

(a) Generic bill of material (GBOM)

For every commercial item (either family or module), there can have one or more commercial module. It may be noted that a commercial family, by definition, cannot be the child in GBOM. A GBOM is illustrated in the Appendix B.

(b) Commercial module-Final product makeup (cmfn-make)



condit	-	condition
co-item	-	commercial item
constra	-	constaint
fami	-	commercial family
fipr-prod	-	final product
modu	-	commercial module
paramete	-	parameter
par-value	-	parameter value
sp-item	-	specific item
stoc-item	-	stock item
stan-prod	-	standard final product
cusp-prod	-	customer specified product
GBOM	-	generic bill of material (gene_bom)
cusp-make	-	customer specified product makeup
cmfp-make	-	commercial family-final product makeup relationship

Fig 3.3 Conceptual schema of BOM for commercial planning and control

A final product can have one or more commercial modules and a commercial module can belong to more than one final products. Thatis, this is a N:M relationship. The need for this relationship is two folds.

(1) The sum of the external prices of all chosen commercial modules is the external price of the commercial product. This is necessary to show the financial consequences of the possible choice (to the customer).

(2) If it is custom specified item, then demand is translated to in terms of specific items (which are MPS-items) and from there to the corresponding planning modules for the calculation of the actual demand.

(c) Condition

This is the relationship between specific-items and parameter values.

(d) Constraints

This details about constraints between different parameter values.

(e) Customer-specified product makeup (cusp-make)

Each customer specified product is made up of at least a stock item. It can be either be a product (which can be a standard final product) or a purchased item.

6. Some issues

The success of the above model crucially depends on following:

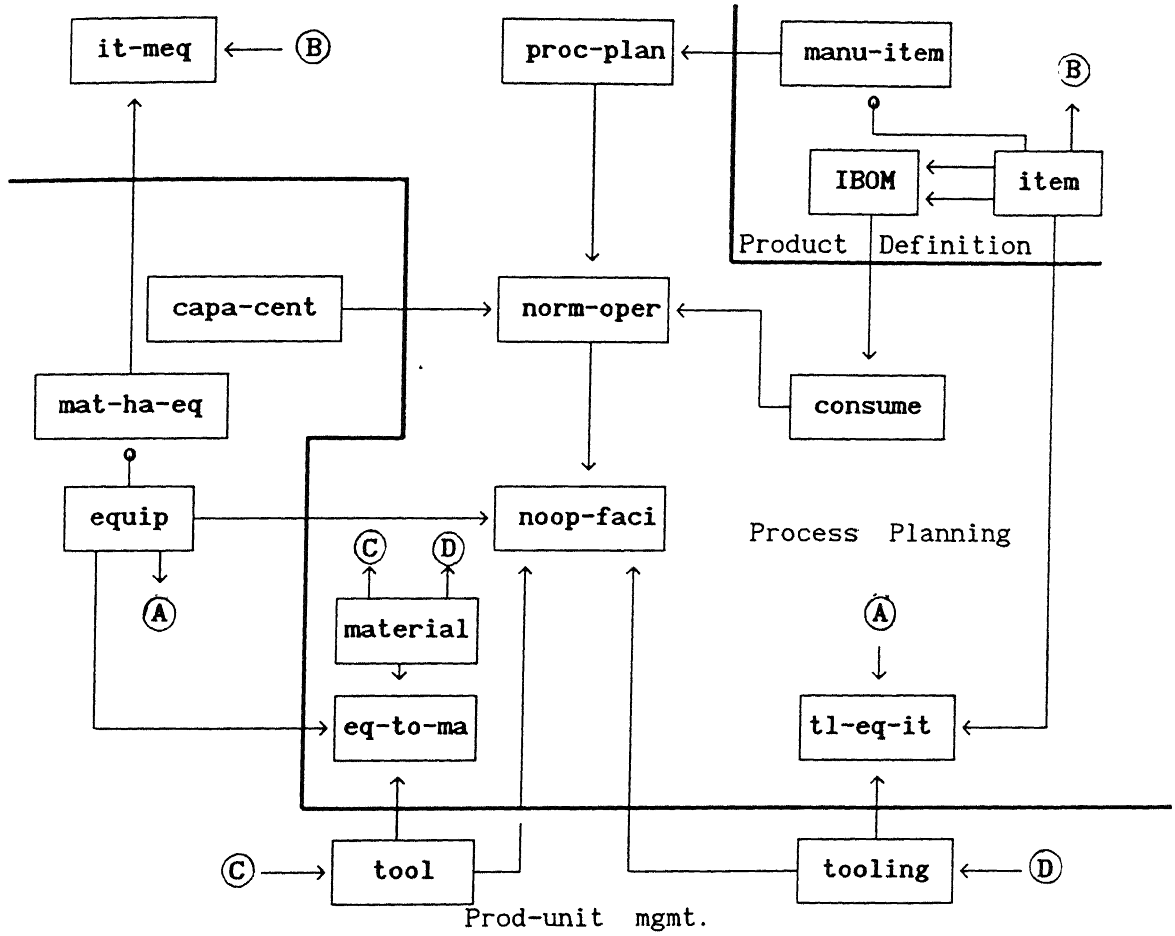
(a) a suitable set of parameters are chosen to characterize products.

(b) the permitted values for these parameters are defined.

(c) dependencies between parameter are determined and recorded.

These must be done in close co-operation with sales and engineering departments so that non-manufacturable products are not offered to customer by sales department, and a manufacturable product could not find a place in the sales departments offer to the customer.

3.3 CONCEPTUAL SCHEMA FOR PROCESS PLAN INFORMATION



proc-plan	- process plan	**manu-item**	- manufactured item
item	- physical item	**capa-cent**	- capacity centre
tool	- machine tool	**tooling**	- tooling item
norm-oper	- normative operation	**material**	- material property
consume	- req. item for normative operation	**IBOM**	- item bill of material (item_bom)
equip	- equipment	**mat-ha-eq**	- material handling equipment

it-meq - item-material handling equipment relationship

noop-faci - normative operation and facility (like tool) relationship

eq-to-ma - equipment-tool-material relationship

tl-eq-it - tooling-equipment-item relationship

Fig 3.4 Conceptual schema of process plan information

1. Entities

The Fig 3.4 shows the conceptual schema for process plan information required for MPC. Each manufacture product may have one or more than one process plans. Each process plan have one or more normative operations.

A normative operation is associated to a capacity center. It is related BOM by an entity called CONSUME. Each normative operation consumes one or more item.

2. Relations

A normative operation facility relationship (noop-faci) shows the association of normative operation with equipment type, tool and toolings.

The three way relationship among the tooling, the equipment and the manufactured item entity relationship contain the tooling applicability knowledge.

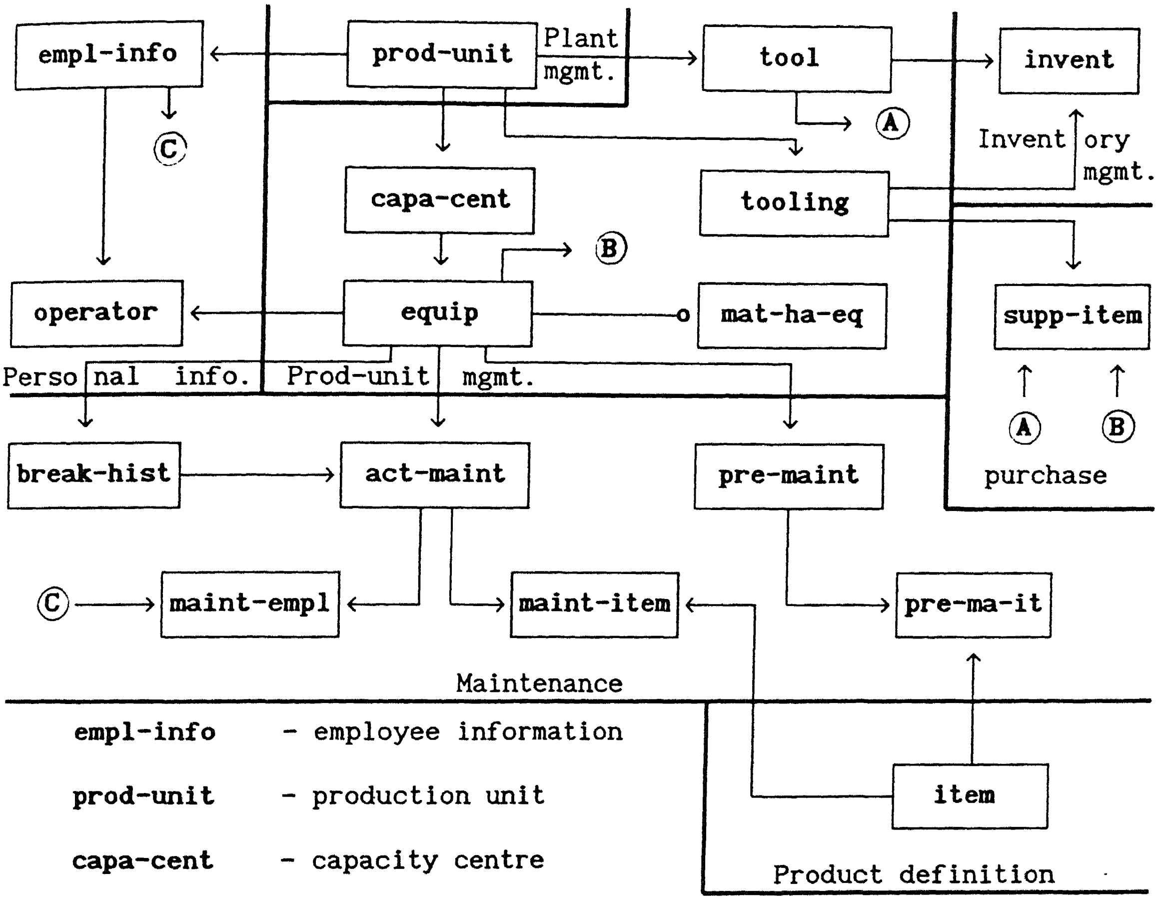
The tool cutting parameters such as feed rates, cutting-speeds, tool life, etc. are the properties of the three way relationship between the tool, the equipment and the material entity relations.

Entity item-material handling contains the information such transfer quantity, standard loading and unloading time, etc. ([26], [27]).

3.4 CONCEPTUAL SCHEMA FOR FACILITY INFORMATION

The Fig 3.5 shows the conceptual schema for facility information required for MPC. Data regarding tools, tooling, equipment, material handling equipments are stored in the respective database tables. An equipment can be a material handling equipment (shown by entity mat-ha-eq).

Each equipment has at least one or more operator. A worker can be the operator of more than one machine. Each person is associated uniquely



empl-info	- employee information
prod-unit	- production unit
capa-cent	- capacity centre
operator	- operator of equipment
equip	- equipment
tool	- machine tool
tooling	- tooling
break-hist	- breakdown history of equipment
pre-maint	- preventive maintenance schedule
pre-ma-it	- preventive maitenance required items
maint-empl	- maintenance employee
maint-item	- maintenance items used
act-maint	- actual maitenance action taken
supp-item	- item and supplier relationship
mat-ha-eq	- material handling equipment
invent	- inventory storage and status

Fig 3.5 Conceptual schema of facility information in production unit

with a production unit.

Each equipment, tool, and tooling can be supplied by one or more suppliers. This relationship is shown in supplied item. Tool and tooling can have one or more storage location (tool and tooling room) and this information is captured in inventory storage and status (invent).

Each machine tool or material handling equipment has a preventive maintenance schedule defined by its builder. This schedule consists of a set of actions. Each action is specified by what maintenance activity needs to be done, when it needs to be done (i.e. after how many hours of machine tool operation) and approximately how long the activity will take. There is a maintenance schedule for each equipment. Actual maintenance actions are performed on each specific equipment on the shop floor. The structure of this information is the same as for the preventive action, but it records date and time of start and end, person did, etc. This maintenance information allows maintenance to be performed so that it does not interfere with the production schedules. The actual maintenance records also the information about breakdowns and repairs (i.e. non-scheduled maintenance). An actual maintenance action can affect one or more equipments [2].

3.5 CONCLUSIONS

The state independent part of the database necessary for MPC is discussed in this chapter. The various product types and the conceptual schema for bill of material for different planning and control functions are detailed. The conceptual schema for process plan information and facility information required for MPC is also developed in the last part of this chapter.

CHAPTER 4

CONCEPTUAL SCHEMA FOR STATE-DEPENDENT PART OF THE DATABASE

The state dependant part of the database stores data regarding the state and transitions of the resources and orders. Materials are received, inspected, stored, consumed by assembling and so on. Customer orders may start as prospects, and are transformed into confirmed orders, completely specified orders, shipped orders, invoiced orders and finished orders. Internal production orders, and orders issued to suppliers and subcontractors have a similar life cycle. All these information-processing takes place at this layer.

The state-dependent part of MPC consists of following functions

(A) - Logistics Planning and Control

Under this, following are included

(1) Order Processing

(2) Demand Analysis Planning (DAP)

(3) Master Production Planning (MPP)

(4) Master Schedule Planning (MSP)

(5) Material Requirement Planning (MRP)

(6) Purchase and Inventory Storage & Transaction

(B) - Production unit (PU) Planning and Control

The PU Planning and Control function consists of

(1) Capacity Requirement Planning

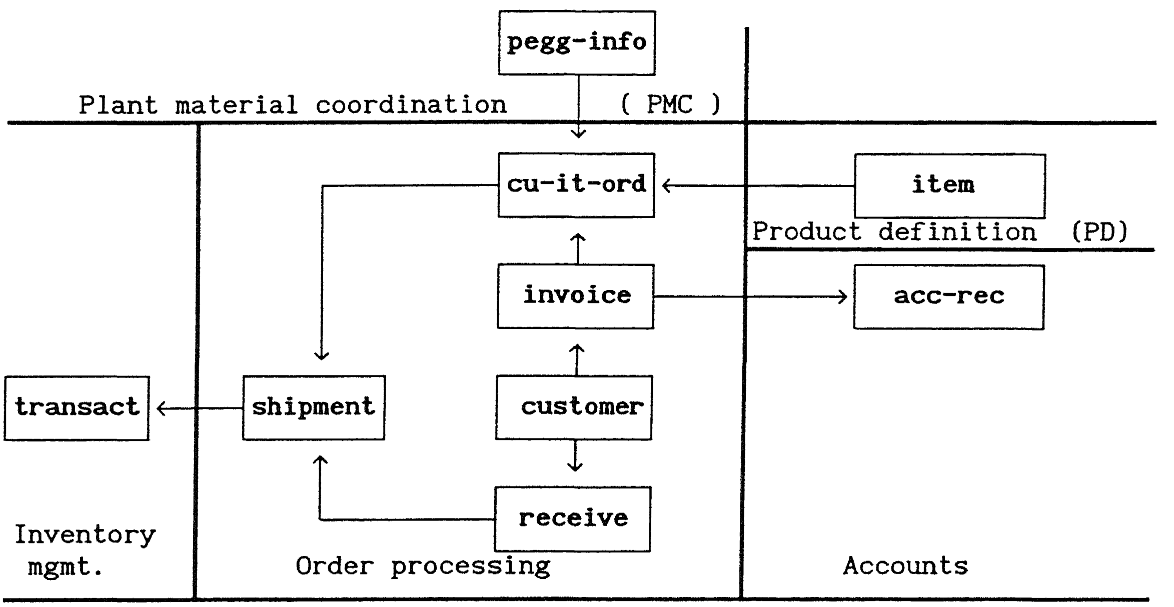
(2) Shop Floor Control (SFC)

This chapter deals with each of the above function and their conceptual schema.

4.1 LOGISTIC PLANNING AND CONTROL

4.1.1 Order processing

The order processing function deals with entry of customer order related activities. The Fig 4.1 shows the conceptual schema for order processing. An order placed by a customer will produce an invoice as well as corresponding customer item orders (cu-it-ord). Each customer item order pertains to an item and can produce one or more shipment. Each shipment entity deals with data regarding the shipment of items ordered by the customer. A customer can have several shipment and can send receiving information for different shipment. A shipment involves at least one inventory transaction from a physical location ([2], [24]).



customer - customer details

acc-rec - account receivable

cu-it-ord - customer item order

pegg-info - pegging information

item - physical item ordered by the customer

transact - inventory transaction details

invoice - invoice (corresponding to the customer order)

shipment - data regarding shipment of items ordered by customer

receive - receiving information send by customer

Fig 4.1 Conceptual schema for order processing

4.1.2 Demand analysis planning (DAP)

Demand analysis planning is the process of identifying anticipated customer demand, period by period, across some horizon of time ([2],[23]).

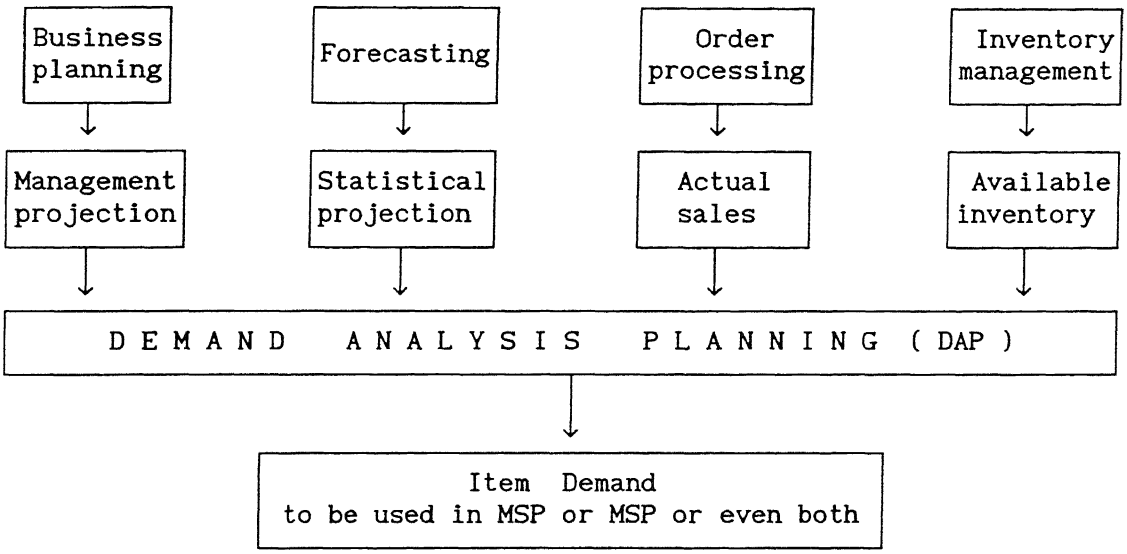


Fig . 4.2 Input and ouput for demand analysis planning

1. Items for planning

Identifying planning modules and items to be planned is a major task, taking into consideration of many aspects. The planning modules (product families) and master products for scheduling (MPS items in MRP(II) literature) are detailed in section 3.1.2.

2. Source of data

The source of data for calculating demand depends on the type of product.For a stable make-to-stock service part, the statistical projection using various forecasting techniques can be the data source. For new product (with no sales history), management projection may be used. For heavily engineered, custom-designed product, probably the manufacturing would start with an actual customer order booking. For product that does have historical sales data, but the statistical projection may be blended with sales data or adjusted with management

projections to have a realistic projection in the light of socio-economic-political environment.

3. Blending

The blending of actual sales to forecast data is done to make the gross demand more realistic. It is done at the item level rather than at product group level. As the Fig 4.3 shows, one can use the forecast data or actual sales or whichever is greater for the gross demand. Mostly some mix or blend of forecast data and actual sales data is done to get the gross demand.

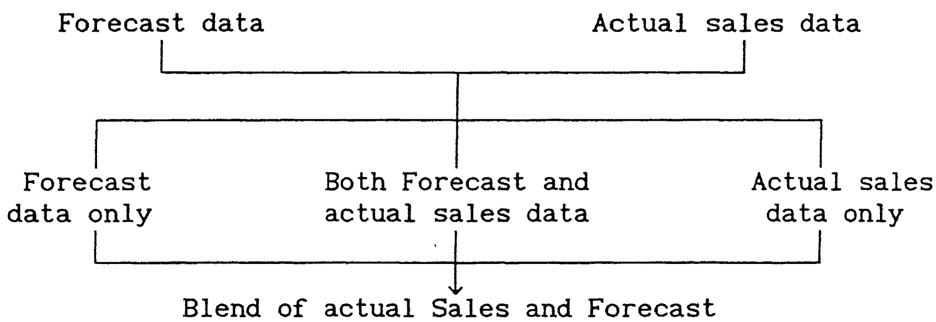


Fig. 4.3 Blending of actual sales to forecast

4. Net demand

The net demand is difference between the gross demand and available inventory of the item.

Available inventory = item on-hand inventory − item planned safety stock − item planned held (hedged) inventory

4.1.3 Master production planning (MPP)

Aggregates the demand for items into product group level. These group or product family demand provide a planner with an 'ease of use' capability and are used for checking against product targets, against available resources, and against the current production plans being used by the firm. ([1], [11]). The Fig 4.4 shows the input and output of MPP.

1. Input

Production targets are established by management as the acceptable or desired production levels for each product group. Sometimes three rates are given. Production minimum implies that the production line will shut down if the actual rate falls below this amount, production desired is the most-effective rate at which to produce the product whereas production maximum is that which can be produced within the available capacity conśtraints.

Item demands are developed by demand analysis.

Current item plans are the item plans developed when MPP was performed on last previous occasion. These are converted to planned orders and scheduled. Now a new MPP plan is constructed and the current plans are used for comparison purposes.

Resource profiles is a set of time offseted data that defines the plant resources necessary to produce a quantity of each MPP item.

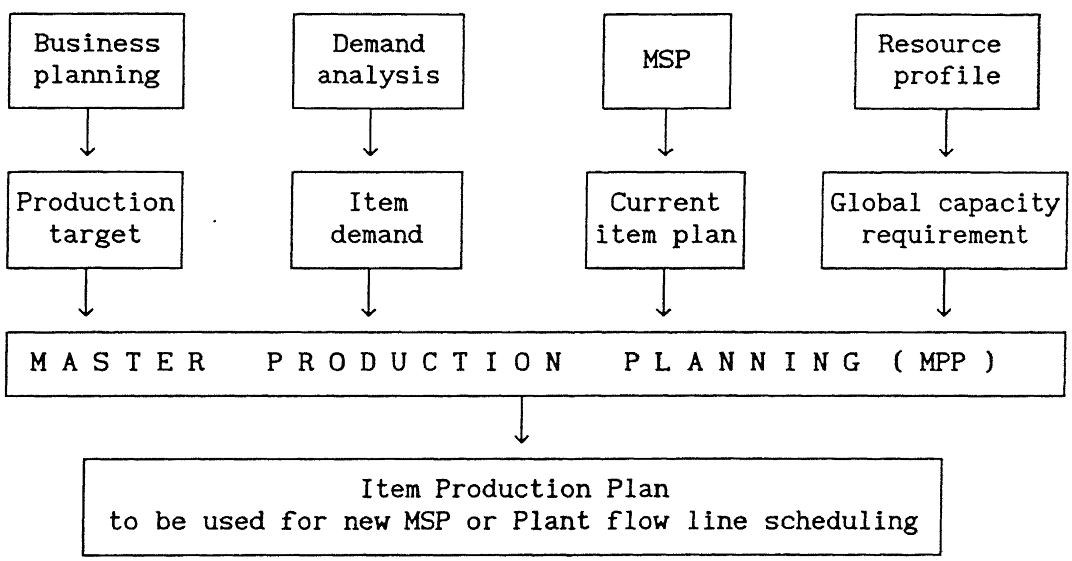


Fig 4.4 Input and output of master production planning (MPP)

2. Output

The output from MPP analysis is either used for directly to schedule a plant flow line or to feed an input for MSP analysis.

4. MPP process

The resource required for the production targets at various point of time is found and is checked to see whether necessary resources are available or not. It can be total labour hours, total machine hours required (at capacity centers), total costs (labour, machine, overhead) etc. and is known as "resource requirement planning (RRP). The intent of RRP is to provide top management with data regarding

(a) the impact of the production plan on the total plant(s) resources at planning time interval or long term period (ie., lt-per and is usually a month or a quarter or half yearly) and for planning time horizon (such as 3 to 5 years)

(b) the major trends of required resources.

If not enough resources available for the target production plan created, top management either has to change production plans or commit additional resource required.

Aggregation of item demands to product families can be based on cost or sales amount or in terms of quantities. However in some firms, such grouping may not be there and hence the planning is done at the item level.

The difference and cumulative difference of product group net demand and production targets at various time interval forms the basis of new production plan. The new plan should meet the net demand for a planning time horizon i.e., cumulative difference is zero (for instance, across three years). Certain periods (lt-per) has a demand spike that cannot be satisfied by that period's production. One option is to lose sales potential.The other is hedging. It is done on those period by building ahead to meet demand in the future. This implies increasing inventory levels and hence most of the firms have hedging time fence which puts a control on how far into the future hedging is allowed. Hedging is

always done at item level rather than product group level. To choose which item in a product group to be hedged depends again on the policy adopted; it can be on the basis of volume or total cost or total sales price, etc.

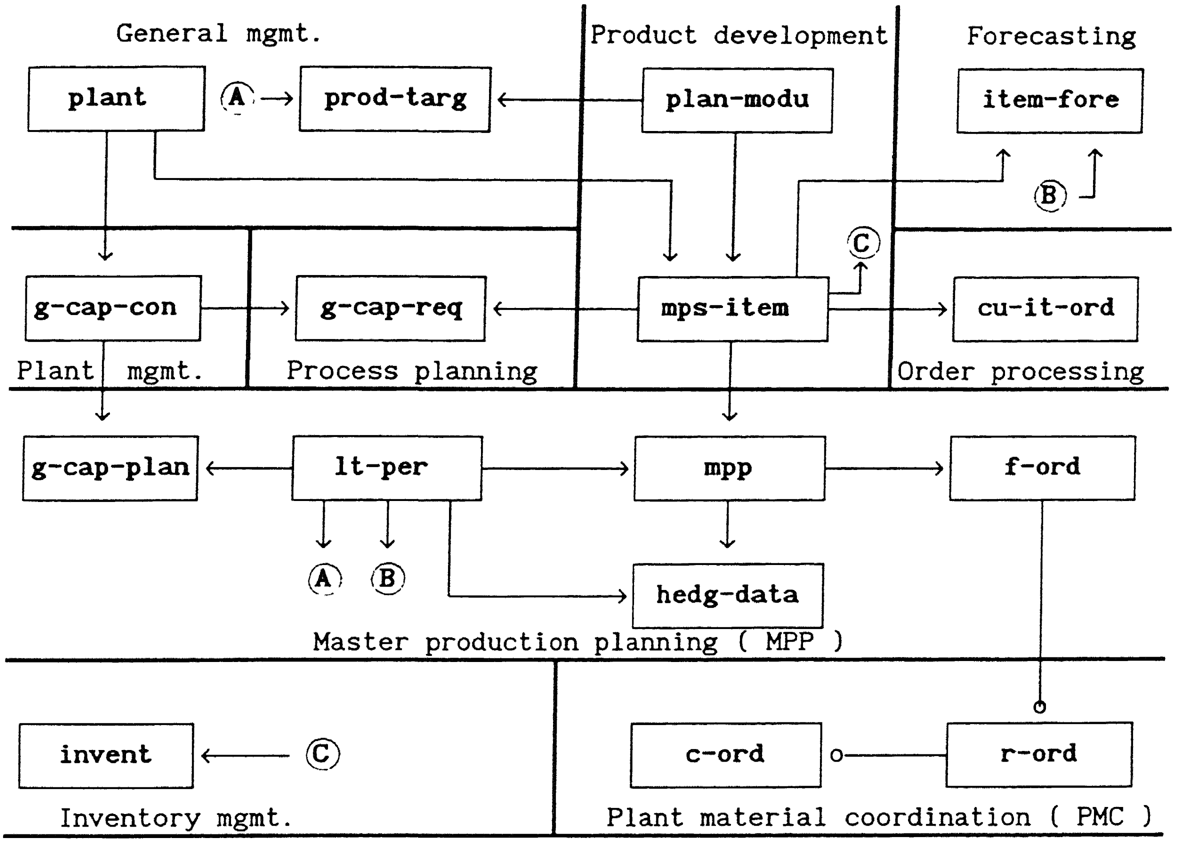
The new production plan is resource checked. Finally it is compared with the existing or operational plan to reduce nervousness (constant change) of the system. When all adjustments have been completed, the item production plans can be released

4.1.4 Conceptual schema for DAP & MPP

The conceptual schema for demand analysis planning (DAP) and master production planning (MPP) is shown in fig 4.5. There can be several plants (under the authority of the general management). Each plant owns a number of global capacity constraints (the entity is g-cap-con). A number of MPS-items and planning modules (or product families) are defined by prouct definition function. The resource profiles of these MPS-items is shown as entity g-cap-req (global capacity requirement). This is used for resource requirement planning and is specified by process planning function.

The entities cu-it-ord (customer item order) from order processing function and item-fore (item forecast) from forecasting function provides information regarding customer order booking and forecast data respectively for the item. The entity invent of inventory storage and transaction function stores data of item on hand inventory for net demand calculation.

The Master Production Planning (MPP) function adds information to this (static) picture. A number of future long-term periods (lt-per) are distinguished and a series of master production plans are created for each items for master planning. Hedging data is retained by the entity hedg-data. The firming up of MPP order belongs to the responsibility of the MPP function. Once orders are firmed, MPP function cannot revoke



plant	- plants of the firm	**invent**	- inventory status and storage
prod-targ	- production target	**hedg-data**	- hedge data
plan-modu	- planning module	**mpp**	- master production plan
item-fore	- item forecast	**mps-item**	- master production scheduling item
lt-per	- long period	**cu-it-ord**	- customer item order
f-ord	- firmed order	**g-cap-con**	- global capacity constraint
r-ord	- released order	**g-cap-req**	- global capacity requirement
c-ord	- completed order	**g-cap-plan**	- global capacity plan

Fig 4.5 Conceptual schema for master planning

without Master Schedule Planning (MSP) function's approval. The figure also shows that the release of orders (r-ord) and the authority to consider an order as completed (c-ord) rests with the MSP function under plant material coordination ([11], [19]).

4.1.5 Master schedule planning (MSP)

While MPP deals with families of items with a large period size bucket (such as a month) across a long horizon (such as three years), MSP deals with individual items, grouped into planned order quantities, with a smaller period size (such as a week) and a smaller time horizon (such as year). This period is called MSP period or medium period (mt-period). For some firms, MPP is not being done. In such situation, MSP would be fed with inputs and does all the process that MPP requires.

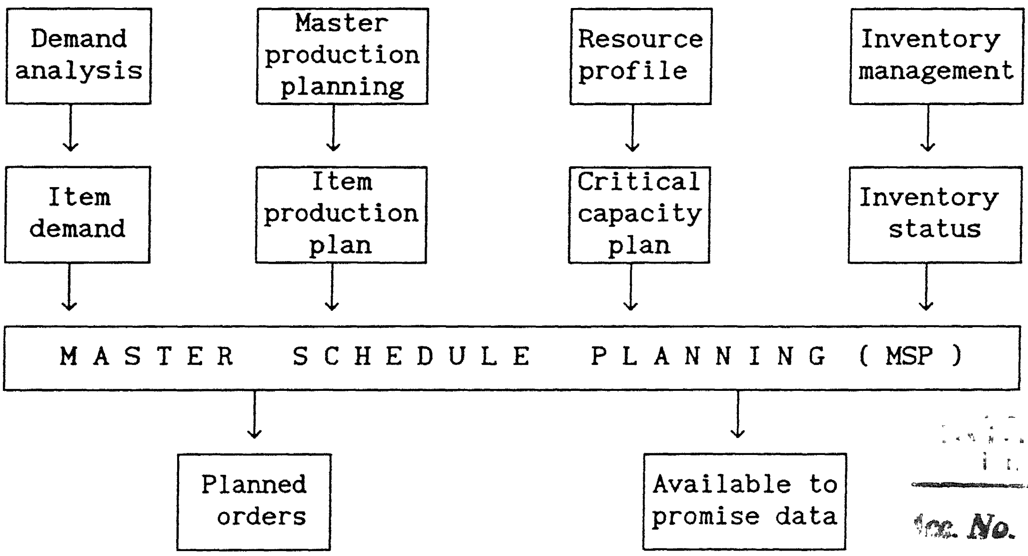


Fig 4.6 Input and output of master schedule planning (MSP)

1. MSP process

MSP accepts the item production plans from MPP and lot sizes them. The lot size is usually based on physical constraints (such as the size of a mixing tank as related to standard batch quantity), economic constraints (such as the cost of a machine set-up), etc. These lot-sized quantities are called "planned orders". These orders are resource checked, usually at

critical centres. This is called Rough Cut Capacity Planning (RCCP). The intent of RCCP is to provide management with the data regarding the impact of lot-sized master schedule orders on selected critical or bottleneck capacity centers at medium or scheduling time interval (usually a week, but can be a day or fortnight) and for scheduling time horizon (such as 6 months to 18 months). The types of data that are of concern at this planning level may be labour hours required, machine hours required.

Usually it is compared with item demands to make more realistic.The available to promise (for customer booking) is an useful data obtained by subtracting successive period required inventory from projected and inventory of that period.

All planned order that crosses "demand time fence" onwards are open orders (released to the shop), becomes a firmed (the planned quantity is not to be recalculated) after CMLT (cumulative material lead time). The planned orders are finally released to MRP ([11], [24]).

4.1.6 Material requirement planning (MRP)

1. MRP logic

The material requirement planning (MRP) derives the time-phased demand of the dependent items from the requirement of the end products and other items that are at the higher level of the product structure using bill of material information and from inventory status. It consists of

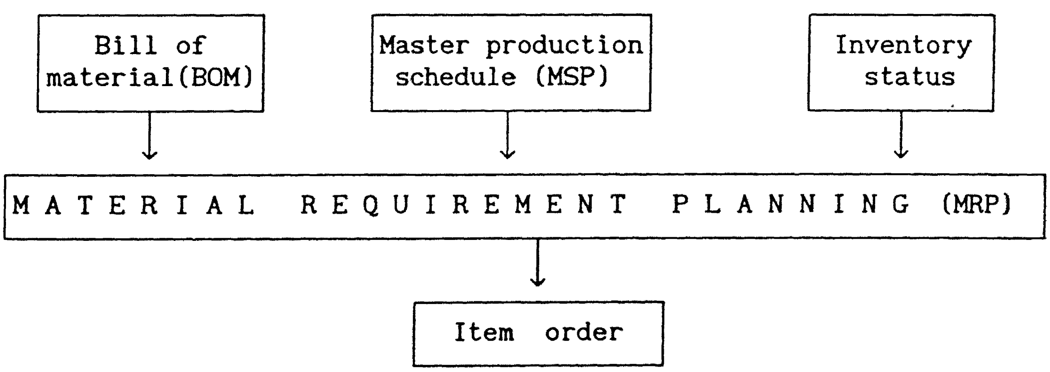


Fig 4.7 Input and output of Material Requirement Planning (MRP)

following steps.

(a) Netting: The process of netting determines how much more of each item (in addition to the available inventory) is needed to satisfy the needs of master schedule. A net requirement for an item occurs when the available inventory of that item is less than the gross requirement,

$$\begin{matrix}\text{Net} \\ \text{req.}\end{matrix} = \begin{matrix}\text{Gross} \\ \text{req.}\end{matrix} - \left(\begin{matrix}\text{Projected on-hand} \\ \text{inventory}\end{matrix} + \text{Scheduled receipts}\right)$$

$$\begin{matrix}\text{Projected} \\ \text{on-hand} \\ \text{inventory}\end{matrix} = \begin{matrix}\text{Physical} \\ \text{on hand} \\ \text{inventory}\end{matrix} - \begin{matrix}\text{allocated} \\ \text{inventory}\end{matrix} - \begin{matrix}\text{hedged} \\ \text{quantity} \\ \text{if any}\end{matrix} - \begin{matrix}\text{safety} \\ \text{stock}\end{matrix}$$

(b) Offsetting: Once net requirements of an item is found, the orders are placed a lead time before needed. This is called offsetting or time-phasing.

c) Exploding: The logic used to compute the quantity all low-level items needed to make a parent is called "exploding". In short,

$$\left.\begin{matrix}\text{The Gross req} \\ \text{for a low-leve} \\ \text{item}\end{matrix}\right\} = \begin{matrix}\text{The net req.} \\ \text{of parents}\end{matrix} \times \begin{matrix}\text{Quantity of low level} \\ \text{item/unit of parent} \\ \text{item}\end{matrix}$$

d) Pegging: Pegging is the reverse process of exploding by which a where-used listing is generated. The where-used listing can be used to trace back what products or assembly cannot be made if some item will not available on time. All requirements of an item for a particular period is stored in the database along with source order (it can be customer order or planned order release). The sum of all such requirement for a period (time-bucket) will be gross requirement of the item for that time bucket is period.

An example to show how MRP is done is given Appendix C.

1. Output

For manufactured item, the output of MRP will be input to CRP,

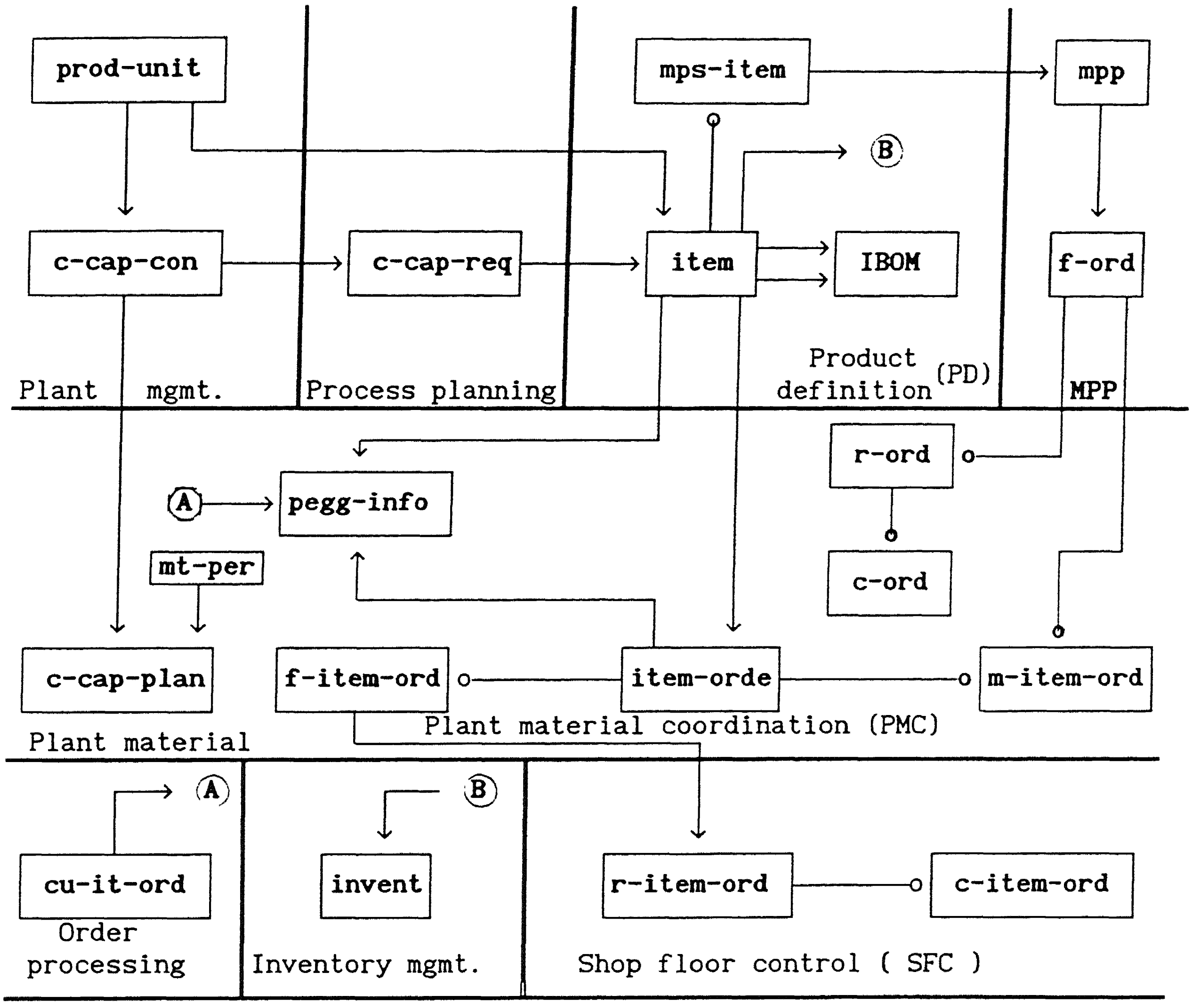
while for purchase item, it is sent for purchase initiation and related activities ([4], [20]).

4.1.7 Conceptual schema for MSP & MRP

The Fig 4.8 shows the conceptual schema for Master Production Schedule (MSP) and Material Requirement Planning (MRP). The is done at plant level and is under the plant material coordination (PMC). The plant management of each plant establishes the production unit (prod-unit) and a number of critical capacity constraints (c-cap-con) per production unit. The process planning function specifies the resource profile i.e., amount of each capacity constraint required (c-cap-req) by each plant item.The items and its bill-of-material (BOM) are specified by product definition function. Some items are MPS-items , and there is always a master production plan specified by MPP function for these items.

The Plant Material Coordination (PMC) function creates a plan of item orders for MPS-items (m-item-ord) of the plant.This is called Plant Master Production Schedule. Based on this, item orders (item-ord) are planned for other items of the plant using MRP logic. These plans are translated into capacity requirement of critical capacity constraints per Production unit, yielding critical capacity loading plans (c-cap-plan) over a number of medium term period (mt-per). This is the Rough Cut Capacity Planning (RCCP). The c-cap-plan is under the responsibility of Plant Material Coordination (PMC) Function. Once an item order is firmed , then PMC function cannot revoke without the concurrence of the concerned PU. Finally the release of item orders and the authority to consider an item order as completed rests with the concerned PU.

The entity pegging relates all the gross requirements for an item to the corresponding sources of demand such as item order (item-ord) or cu-it-ord (customer item order). Once an order is released, the required

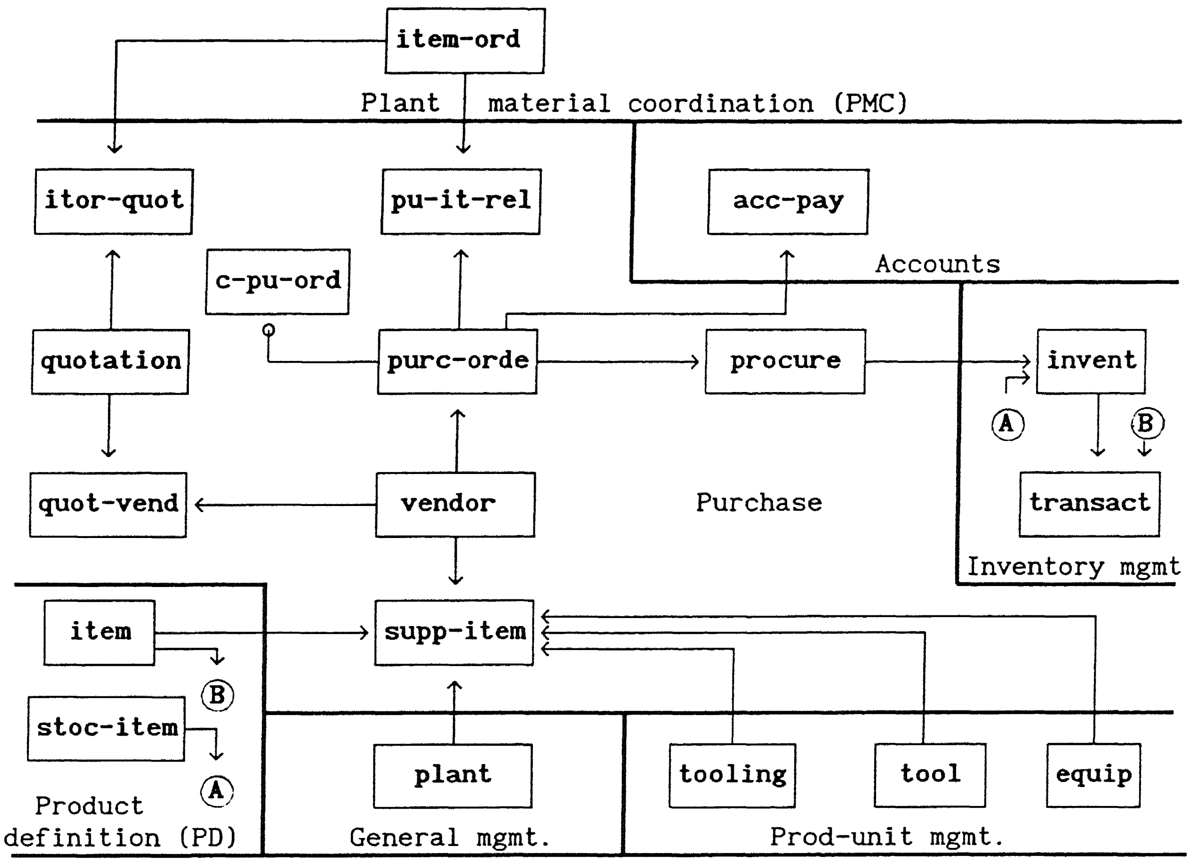


prod-unit	- production unit	**c-cap-con**	- critical capacity constraints
mps-item	- master production scheduling item	**c-cap-req**	- critical capacity requirement
mpp	- master production plan	**c-cap-plan**	- critical capacity plan
f-ord	- firmed order	**IBOM**	- item bill of material
r-ord	- released order	**manu-item**	- manufactured item
c-ord	- completed order	**item**	- items of the plant
item-orde	- item order	**mt-per**	- medium term period
m-item-ord	- mps item order	**pegg-info**	- pegging information
f-item-ord	- firmed item order	**cu-it-ord**	- customer item order
r-item-ord	- released item order	**invent**	inventory status and storage
c-item-ord	- completed item order		

Fig 4.8 Conceptual schema for plant material coordination (PMC)

quantity of items or material necessary for its production is allocated and thus each shop order is 'pegged' to the allocated quantity in the pegging entity ([4], [19], [20]).

4.1.8 Purchase and inventory storage and transaction



item-ord	- item order	**quot-vend**	- quotation and vendor relationship
itor-quot	- item order and quotation relationship	**vendor**	- supplier information
pu-it-rel	- purchase order and item order relationship	**transact**	- inventory transaction details
acc-pay	- account payable	**item**	- items purchased
c-pu-ord	- completed purchase order	**stoc-item**	- items stocked
quotation	- quotation called	**supp-item**	- item and supplier relationship
purc-orde	- purchase order	**tool**	- tools used for manufacture
procure	- procurement of purchased item	**tooling**	- tooling required
invent	- inventory status and storage info.	**equip**	- equipment of the plant

Fig 4.9 Conceptual schema for Purchase function

All purchase item order with lead to one or more purchase release orders. A purchase item order may have one or more quotation. A purchase release order should have at least one purchase item order and is precisely given to a vendor.

Each purchase release order will have an entity called procurement. It stores the information which includes the mode of shipment, the physical location where the item is to be sent, inspection document, etc. Once a purchase release order is closed, its details are summarized in closed purchase order (cu-pu-ord).

Every movement of item from a physical location involves a inventory transaction, identified by pick ticket no. (Pick #)

4.2 PRODUCTION UNIT (PU) PLANNING AND CONTROL

4.2.1 Capacity requirement planning (CRP)

1. CRP process

CRP deals with "now". It is accomplished to determine if capacity exists to produce the planned items at a work center, then at what shift, which day or which week, it should be done. The detailed MRP data establish the exact order quantities and timings (planned order of item; item release order) for use in calculating the capacity required. The resultant capacity needs are summarized by time period and by capacity center in a format similar to that time phased planned order of MRP except it is late by the offset (as established in resource profile of the item).It would be a accurate projection of capacity center requirements ([3], [23]). Appendix D gives an example to show how CRP data is generated.

2. Database requirement

CRP requires no additional database other than that is required for

MRP, but needs additional computer run time. However, several other factors influence the design and the maintenance of the database for CRP. The level of detail appropriate for capacity management implies a corresponding level of detail in the database and in database maintenance. If the capacity assessment are made in terms of sales rupees or average labour hrs, the data may be extracted from the financial accounting database. Secondly, incorporating what-if-analysis into the capacity planning system in order to evaluate different material plans implies additional demand on the database design. Care must be taken to isolate the actual MPS and MRP records from changes made by what-if-analysis [24].

4.2.2 Shop floor control

This function consists of 3 steps.

1. Order release

The purpose of the order release is to provide the necessary documentation that accompanies a shop order as it is processed through the shop.Following are documents generated by this function

(a) Route sheet - listing the operation sequence and tools needed

(b) Material requisitions - to draw necessary raw materials (or components for assemblies) from stock

(c) Job cards - to issue to the labour

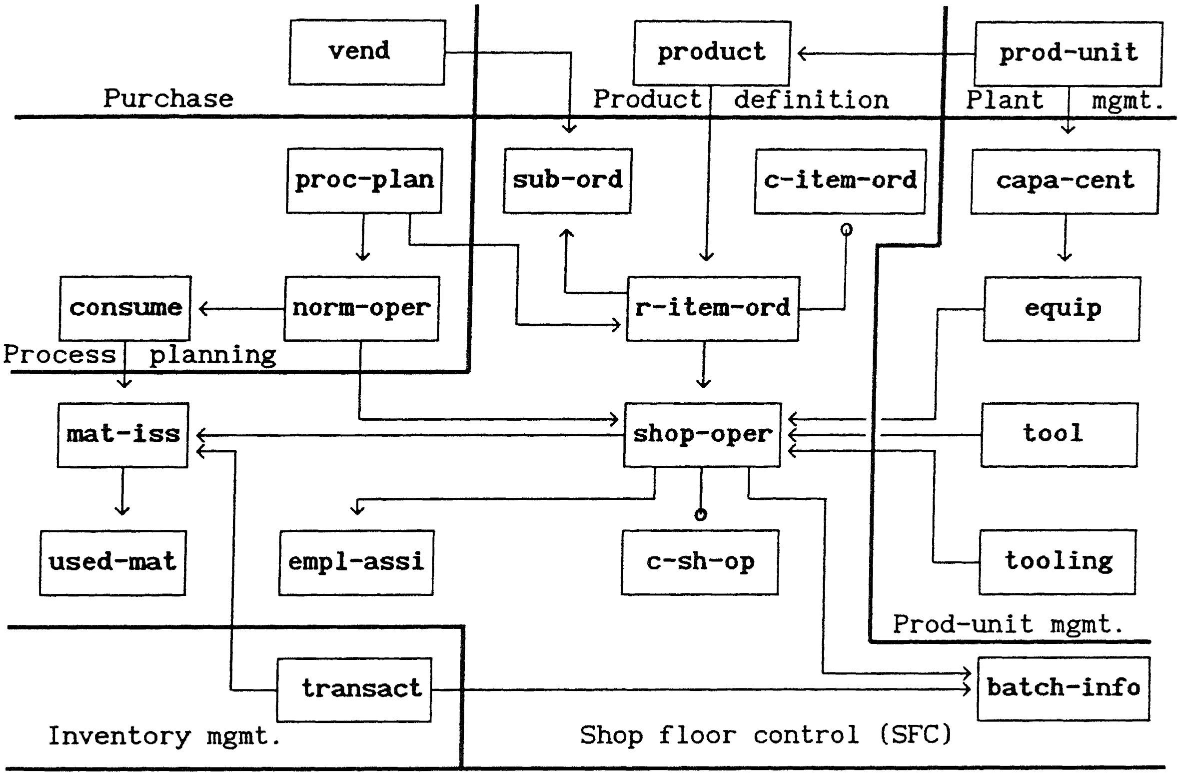
(d) Move tickets - to move the parts between work centers

(e) Parts list - assembly jobs

The two inputs required are (i) item order (or work order) and (ii) the standard routing data and bill of material data.

2. Order scheduling

The purpose of scheduling is to make assignments to the various machines in the factory. Inputs to this function consist of the order release data and priority control (determined from its due date and other



product	- manufactured product	**shop-oper**	- shop operation
proc-plan	- process plan	**used-mat**	- material used for shop operation
sub-ord	- subcontractor order	**empl-assi**	- employee assignment
r-item-ord	- released item order	**c-sh-op**	- completed shop operation
c-item-ord	- completed item order	**batch-info**	- manufactured product details
capa-cent	- capacity center	**tool**	- tool used in the shop
consume	- items req. for normative operation	**tooling**	- tooling used in the shop
norm-oper	- normative operation	**equip**	- equipment used in the shop
mat-iss	- material issued	**prod-unit**	- production unit
vendor	- vendor for subcontracting	**transact**	- Inventory transaction

Fig 4.10 Conceptual schema for shop floor control (SFC)

factors). The basic document produced by order schedule function is the dispatch list which reports the jobs that should done at each capacity center and certain details about the routing of the part

3. Order progress

This function monitors the progress of the released shop orders and collect data from shop floor.The data stores in the open shop order files.

4.2.3 Conceptual schema for PU planning and control

For each production unit, the conceptual schema is shown in the fig 4.10. It shows that for each PU consists of a number of capacity centers (capa-cent), where there can be no of equipments (equip). Each PU may also be allocated with tools and toolings required for manufacturing.

For each item (of the plant), there are one or more process plans (proc-plan) for which there are number of normative operations. Each normative operation also consumes certain items as input (consume).

Each firmed item order (f-item-ord) from PMC function will be released (r-work-ord), with a process plan associated with it. The dispatching of shop orders and the related material issue (mat-iss) and employee assignment (empl-assi), associated with released work order, are under the responsibility of shop floor control function (SFC). The SFC function is also responsible to consider a shop order as completed and register the related data. The finished product data is stored in the entity batch-info.

4.3 CONCLUSIONS

The present chapter deals with the state dependant part of the database required for MPC. The modular conceptual schema for the three levels of manufacturing planning and control is developed. This chapter also details the schema for the associate functions of the MPC such as order processing and purchase.

CHAPTER - 5

IMPLEMENTATION

This chapter discusses about the implementation of database based on the conceptual schema developed in the third and fourth chapters. The present software is developed in the relational database management system (RDBMS), INGRES version 6.2/03 hp installed in the main frame HP 9000/800 in IIT-K. The application is developed in public database DEMO. The construction of the software is explained using the frame structure which navigates to all the constituents of the system. The Fig 5.1 shows the main frame. All frames have menu for accessing the next frames or database tables. It also have menu for quit and is not shown in the Figures. Once a database table is accessed, there are menus for creating, editing or deleting the various entries (to be more precise, various entities). These are not shown in the Figures. Appendix E lists the various database tables for each of the entities, with its owner function and the attributes. Appendix F lists the various forms, frames and programmes.

5.1 MAIN FRAME

DATABASE FOR MANUFACTURING PLANNING AND CONTROL	
1. MASTER PLANNING	MAST
2. PLANT MATERIAL COORDINATION	MACO
3. PRODUCTION UNIT CONTROL	PUCO
4. QUIT	QUIT

Fig. 5.1 MAIN FRAME

All database tables are organized under three major category. In the main menu, the option 1 or menu MAST takes to the frame for choosing database tables under MASTER PLANNING. The option 2 or menu MACO takes the user to the database tables under PLANT MATERIAL COORDINATION, while option 3 or menu PUCO leads to the frame for selecting the database tables under PRODUCTION UNIT CONTROL. The option 4 or QUIT makes the user to log out from the system.

5.2 MASTER PLANNING

DATABASE FOR MASTER PLANNING	
1. PLANT INFORMATION	PLANT
2. PRODUCTION TARGET	PRTA
3. ITEM FORECAST	ITFO
4. ORDER PROCESSING	ORPR
5. MASTER PRODUCTION PLANNING	MPP
6. PURCHASE	PURC
7. BILL OF MATERIAL	BOFM

Fig. 5.2 Frame for choosing database tables under master planning

The first option in the main frame lets the user to enter the database for master planning. The Fig.5.2 shows the various database tables that one can access under this option. It includes plant information, production targets, item forecast, and various database tables under order processing, master production planning, purchase and bill of material. Under option for order processing, following database tables can be accessed: customer information, invoice preparation, shipment, receiving information and customer item order. This is shown in the Fig.5.3. Similarly, Figures 5.4 and 5.5 lists the various database tables that can be accessed, with corresponding menu, under option for

production planning and purchase respectively.

The option 7 or menu BOFM, as shown in the Fig.5.2, takes the user to the database tables for bill of material. It is organized under three heads and is shown in Fig.5.6. These three heads are BOM for commercial planning and control (option 1), BOM for logistic planning and control (option 2) and BOM for production unit planning and control (option 3).

DATABASE FOR BILL OF MATERIAL (BOM)	
1.	COMMERCIAL PLANNING AND CONTROL
2.	LOGISTICS PLANNING AND CONTROL
3.	PRODUCTION PLANNING AND CONTROL

Fig.5.6 Frame for choosing database tables under bill of material

DATABASE OF BOM FOR COMMERCIAL PLANNING AND CONTROL	
1. COMMERCIAL ITEM	COIT
2. CONDITION	COND
3. CONSTRAINT	CONS
4. FINAL PRODUCT	FIPR
5. GENERIC BILL OF MATERIAL	GBOM
6. PARAMETER	PARA
7. SPECIFIC ITEM	SPIM
8. COMMERCIAL MODULE & FINAL PRODUCT RELATIONSHIP	CMFP

Fig.5.7. Frame for choosing database tables of BOM under commercial planning and control

Figures 5.7, 5.8 and 5.9 show the various database tables of BOM that can be accessed under commercial planning and control, logistic planning and control and PU planning and control respectively.

DATABASE OF BOM FOR LOGISTIC PLANNING AND CONTROL	
1. PLANNING MODULE	PLMO
2. COMMERCIAL FAMILY AND PLANNING MODULE RELATIONSHIP	CFPM
3. PLANNING MODULE AND STANDARD FINAL PRODUCT RELATIONSHIP	PMSP
4. MASTER PRODUCTS FOR SCHEDULING	MPSI

Fig.5.8. Frame for choosing database tables of BOM under logistic planning and control

DATABASE OF BOM FOR PRODUCTION UNIT PLANNING AND CONTROL	
1. ITEM	ITEM
2. ITEM BILL OF MATERIAL	IBOM

Fig.5.9 Frame for choosing database tables of BOM under production unit planning and control

5.3 PLANT MATERIAL COORDINATION (PMC)

The second option in the main frame takes the user to the frame for choosing database tables under PMC and is shown in the Fig.5.10. It consists of database tables for production unit information, global

DATABASE FOR PLANT MATERIAL COORDINATION	
1. PRODUCTION UNIT INFORMATION	PRUN
2. MATERIAL COORDINATION	MCOP
3. PROCESS PLAN INFORMATION	PRPL
4. INVENTORY STORAGE AND TRANSACTION	ISTT
5. GLOBAL CAPACITY CONSTRAINT	GCAC
6. CRITICAL CAPACITY CONSTRAINT	CCAC

Fig.5.10 Frame for choosing database tables under plant material coordination

capacity constraint, critical capacity constraint and various databases

tables organized under material coordination, process plan information and inventory storage and transaction.

The Figures 5.11, 5.12 and 5.13 show frames for selecting the various database tables, with corresponding menus, that can be accessed from the frames for database for material coordination, inventory storage and transaction and process plan information respectively.

DATABASE FOR MATERIAL COORDINATION	
1. RELEASED ORDER	REOR
2. COMPLETED ORDER	COOR
3. PEGGING INFORMATION	PEGG
4. ITEM ORDER	ITOR
5. MPS ITEM ORDER	MIOR
6. FIRMED ITEM ORDER	FIOR
7. MEDIUM TERM PERIOD	MTPR
8. CRITICAL CAPACITY PLAN	CCPL

Fig.5.11 Frame for choosing database tables under material coordination

DATABASE FOR INVENTORY STORAGE AND TRANSACTION	
1. INVENTORY STORAGE AND PHYSICAL LOCATION	INVENT
2. INVENTORY TRANSACTION	TRANS

Fig.5.12 Frame for choosing database tables under inventory mgmt

DATABASE FOR PROCESS PLAN INFORMATION	
1. PROCESS PLAN	PROP
2. NORMATIVE OPERATION	NOOP
3. EQUIPMENT-TOOL-MATERIAL RELATIONSHIP	EQTOMA
4. EQUIPMENT-TOOLING-ITEM RELATIONSHIP	EQTLIT
5. ITEM-MATERIAL HANDLING RELATIONSHIP	ITMEQ
6. NORMATIVE OPERATION AND FACILITY RELATIONSHIP	NOFA
7. CONSUME	CONE
8. SUBCONTRACT REQUIREMENT	SURE
9. GLOBAL CAPACITY REQUIREMENT	GCAR
10. CRITICAL CAPACITY REQUIREMENT	CCAR

Fig.5.13 Frame for choosing database tables under process plan information

5.4 PRODUCTION UNIT CONTROL

The third option in the main frame takes the user to the database for production unit control. The various database tables coming under this section is grouped in three divisions and is shown in the Fig.5.14. The

DATABASE FOR PRODUCTION UNIT CONTROL	
1. FACILITY INFORMATION	FACI
2. MAINTENANCE	MAINT
3. SHOP FLOOR CONTROL	SFC

Fig.5.14 Frame for choosing the options under production unit control

option for facility information takes the user to the frame for database for facility information. The Fig.5.16 lists the various database tables that can be accessed under this option. Similarly Figures 5.16 and 5.17 show the various database tables that one can access under the option for maintenance and shop floor control respectively.

DATABASE FOR FACILITY INFORMATION	
1. TOOL INFORMATION	TOIN
2. TOOLING INFORMATION	TLIN
3. EQUIPMENT INFORMATION	EQIN
4. MATERIAL HANDLING EQUIPMENT INFORMATION	MHEI
5. CAPACITY CENTRE INFORMATION	CACE

Fig.5.15 Frame for choosing database tables under facility information

DATABASE FOR MAINTENANCE	
1. MAINTENANCE EMPLOYEE	MAEM
2. MAINTENANCE ITEM	MAIT
3. BREAKDOWN HISTORY	BREAK
4. ACTUAL MAINTENANCE	ACMA
5. PREVENTIVE MAINTENANCE ACTION	PRMA
6. PREVENTIVE MAINTENANCE ITEM	PRIT

Fig.5.16 Frame for choosing database tables under maintenance

DATABASE FOR SHOP FLOOR CONTROL	
1. RELEASED ITEM ORDER	RIOR
2. COMPLETED ITEM ORDER	CIOR
3. SUBCONTRACT ORDER	SORD
4. SHOP OPERATION	SHOP
5. COMPLETED SHOP OPERATION	CSHO
6. EMPLOYEE ASSIGNMENT	EMAS
7. MATERIAL ISSUED	MAIS
8. USED MATERIAL	USMA
9. BATCH INFORMATION	BAIN

Fig.5.17 Frame for choosing database tables under shop floor control

CHAPTER - 6

CONCLUSIONS

6.1 CONCLUSIONS

In the present work, the design and the development of database for manufacturing planning and control (MPC) in a make to stock manufacturing environment is considered and a software is developed for the same.

The problems of having common database and several stand alone databases are looked into and suggested decomposition at the organizational and conceptual level as a solution to this. Based on this, modular conceptual schema for state-independent as well as for state-dependant part of the database are developed. The owner function of each of these modules are also identified.

The system that is developed group database tables for various entities according to the functions that own them. It also allows the user to create, delete, retrieve and update the entries (or entities, to more precise) in the database tables.

6.2 SCOPE FOR FUTURE STUDY

Although quite a good time was spent in identifying entities and collecting their data elements, more attributes could be incorporated to each entities that are relevant to a specific situation.

The present work could be extended for a make to order manufacturing environment on a similar line.

The database does not accommodate the geometry features of the product. A study can be made to include these aspects also.

Even though effort is placed in linking the various database tables, still refinement can be done to make it more integral system. The transformation as well a as dynamic constraint over data as it moves from one stage to another could not be implemented, although it requires some specific situations to be considered. The use powerful language like C can be considered to achieve such a interface.

REFERENCES

(1) Bertrand; Wortmann; Wijngaard .,"Production control - A Structural and Design Oriented Approach", Elsevier, 1990.

(2) Bray, O.H., "CIM The Data Management Strategy" , Digital Press, 1988.

(3) Browne, J.,"Production activity control - a key aspect of production control" , Int. J. Prod. Res. 1988, Vol. 26, No.3, pp. 415-427.

(4) Chu, Chao-Hsien; Nilakanta, Sree .,"On the design of a micro-based MRP system a relational database approach" .,Computers & Industrial Engineering, Vol. 15, No. 1-4, pp. 146-152, 1988.

(5) Cohen, Oded, "The Drum-Buffer-Rope (DBR) Approach to Logistics ., IFIP State-of-the Art Reports, Edited by A. Rolastads 1988.

(6) Course notes on Igres .,Mastek Pvt. Ltd.

(7) Date, C.J., "A Guide to Ingress" .,Addison-wesley Pub. Com., 1987.

(8) Date, J., "An Introduction to Database Systems" ., Third Edition, Addision-wesley Pvt. Com. 1981.

(9) De Brock, E.O.; Remmen, F. Wortmann, J.C., "A Conceptual Model for Product-types and Bills of material: Application of set-theory Approach " , proceedings of the IFIP TCS/WG 5.7 - Databases for Production Management , North holland , 1990

(10) Galbraith,J ., "Designing Complex organizations" ., Addison-Wesley, Reading Mass ., 1973

(11) Gessener, R., "Master Production Schedule Planning", Wiley & Sons, 1986.

(12) Goldratt, Eliyahu, In "Computerized Shop Floor Scheduling" Int. J. Prod. Res. 1988, Vol. 26, No.3, pp. 443-455.

(13) Groover,Mikell.P, Zimmers, Emory.W Jr ., "CAD/CAM Computer aided design and manufacturing , PHI , 1987

(14) Harhen, John, "MRP/MRP II", IFIP State-of-the-art-reports edited by A. Rolstadas, 1988.

(15) Meal, Harlan, C., "Putting Production Decisions where they belong", Harvward Business Review, Vol. 84, No.2, 1984, pp. 102-111.

(16) Morgan, Rachel; Mc Gilton, Henry, "Introducing UNIX System V", McGraw Hill, 1987.

(17) Orlicky, J., "MRP Material Requirement Planning, The New Way of Life in Production and Inventory Management", McGraw-Hill, N.Y, 1973

(18) Pels,H.J ., "Conceptual Integration of Distribuited PMS Databases, proceedings of the IFIP TCS/WG 5.7 - Databases for Production Management , North holland , 1990

(19) Pels, H.J.; Wortmann, J.C., "Decomposition of Information Systems for Production Management", Computers in Industry, 6(1985) 435-452.

(20) Regan, S.; Harhen, J.; Browne, J.; O'Kelly, M.E.J.; "The Design of a Microcomputer-Based Net Change MRP System".

(21) Romeo, A.; Esparrgo, Jr., "Kanban"; Production and Inventory Management Journal, First quarter, 1980, pp. 6-10.

(22) Scheer, A.W.; "CIM Computers Steered Industry", Springer Verlag,1988.

(23) Veen, E.A. Van; Wortmann, J.C.; "Generic Bills of Material in Assemble-to-order Manufacturing", Int. J. Prod. res. 1987, Vol. 25, No.11, pp. 1645-1658.

(24) Vollmann, T.E.; Berry, T.E.; Whybark, D.C., "Manufacturing Planning and Control Systems", Galgotia Publications Pvt. Ltd., 1989.

(25) Rao, Ashok; Sahergy, David, "Moving from Manufacturing Resource Planning to just-in-time Manufacturing". Production and Inventory Management Journal first quarter, 1988, pp. 44-49.

(26) Wang,Ming, Smith.G.W., "The Intelligent Database Design Environment (IDDE) for the Design and Implementation of CIM information system" , proceedings of the IFIP TCS/WG 5.7 - Databases for Production Management , North holland , 1990

(27) Wang, Ming; Walker, H., "Creation of an Intelligent Process Planning System within the Relational DBMS Software Environment" Computers in Industry (13) (1980), pp. 215-228.

(28) Yamoshina, Hajime, "Just-in-Time Production - A New Formulation and Algorithm of the Flow Shop Problem", IFIP State-of-the art-reports edited by A. Rolstadas, 1988.

APPENDIX A

CONCEPTUAL SCHEMA (CONCEPTUAL DATA MODEL)

1. Some terminologies

A possible content of the database is referred to as database state. A database state is a structured collection of sentences. A sentence refers to one or more entities and facts about these entities in the universe of discourse. A tuple in a relational database is an example of a sentence with a formalized structure. A set of equally structured sentences is called a table (see appendix E for various tables of the present implementation). The set of all allowed database states (corresponding to the set of possible and distinguishable states of the universe of discourse) is called database universe. A transition is a change of database state. It can be due insertion, deletion or update.

2. Constraint

Not all possible database state are meaningful. So a number of so called static integrity constraints must be defined in the conceptual schema to specify precisely which database states are allowed. Key and referential constraints are typical examples of static constraints.

Dynamic constraints restrict the set of allowed database transitions. For instance, if a data element, say salary, cannot decrease, but only increase, then it forms a dynamic constraint.

3. Elements of a data model

(a) Entity type and attributes : Entity types represent the basic concepts in a universe of discourse. Attributes represent the properties that characterize the concept. A key is a subset of attributes of an

entity type that uniquely identifies each occurrence of that entity type.

(b) Referential constraints : If X and Y are entity types, Ax is a key of X, Ay is a subset of attributes of Y, x is an occurrence of X and y is an occurrence of Y, then a referential constraint on Ay with respect to Ax requires that for each y, the attributes of Ay have values that refer to equal values of corresponding attributes in Ax of X. Because Ax is a key of X, a referential constraint specifies a one to many relationship from X to Y : x can be related to zero or more occurrences of Y and y is related to exactly an occurrence of X. Here X is the owner and Y is the member entity of the referential constraint (Bachman convention). It is indicated with an arrow from the owner to the member entity [19].

(c) Generalization constraint : This is a special case of a referential constraint, where not only the subset of attributes Ax is a key of the owner entity type, but also Ay is a key of the member entity type. This implies that a generalization constraint establishes a one to one relationship from the owner X to member Y : every y is related to precisely one x, every x is related zero or one y. As a result Y can be regarded as a special case of X, that has some special attributes and be called subentity of X. X is the general case that has only the general attributes. Generalization constraint is indicated in the Bachman diagram by an arc with a '0' instead of arrow.

4. Relational model

In the relational data model, the data structure involves

relations, domains and tuples and the basic operations are selection, projection and join. A relation is simply a table of data with rows and columns. The data about a specific type of entity are selected in one or more relations. The attributes of the entity are stored in the columns of the relation and are called domains. Each row in a relation represents a specific occurrence of that type of entity and is known as tuple.

The operation selection involves 'cutting out' or specifying one or more row or tuples in a relation having certain specific values of the attribute that is specified. With projection, only those attributes that are specific with operation is sliced out. Thus with operations selection and projection, any data element in the table can be assessed. The join operation involves combining data from two different relations where there is at least one common domain or column.

5. Primitive operation

The inclusion of an entity or relationship into the set of entities or relationships of the same type is called insertion, while its removal or deletion is the operation deletion. The change of values of attribute of an entity or relationship is known as update. Strictly speaking an update is a combination of deletion and insertion.

6. Normalization

Normalization ensures that the relational model will be designed in such a way that the database eventually derived will not contain redundant data and cause operation anomalies (such as during insertion, deletion, update), thus maintain data integrity. It is a process of nonloss decomposition in which a given relation can be decomposed into the elementary forms and, the join operation can be applied over them to produce the original relation without losing information. Normalization is often described in five successive stages, with certain undesirable

features are eliminated from unnormalised relation after each stage.The classification of attributes into key and non-key attributes is according to the functional dependency, which could be (a) full or partial, (b) transitive or non-transitive. For a composite key attribute, there are two possible types of dependencies among its component attributes, ie multi-valued dependency and join dependency. The five stages are as follows

Obtain 1NF relations by eliminating
repeating group

↓

Obtain 2NF relations by eliminating
nonfull functional dependencies

↓

Obtain 3NF relation by eliminating
transitive dependencies

↓

Obtain 4NF relation by eliminating
any multi-valued dependencies that are not functionally dependant

↓

Obtain 5NF relation by eliminating
any join dependencies that are not results of the key.

7. Integrity rules

The integrity rules aims to ensure that, when attribute values or tuples in a relation are inserted or deleted, the transformation will not lead the database from a consistent state into an inconsistent state.

(a) Entity integrity rule : none of a primary key values be null.

(b) Relationship rule or referential integrity rule : an occurrence of a relationship set may exist in the database only if the occurrence of the entities participating in the association also exist in the database.

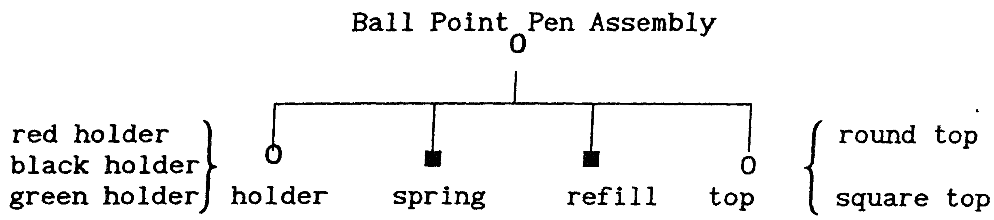
(c) Sub entity integrity rule : for every sub entity set E_i ($i = 1,..,n$) in the database, there must exist its generic entity set E, and, on the other hand, given any sub entity set E_i, all the properties of its generic entity set are applicable to E_i.

APPENDIX B

GENERIC ITEM AND GENERIC BILL OF MATERIAL

An example to highlight generic items, parameter and parameter value with conditions and constraints is given here.

The multi level generic BOM of a ball point pen with generic items at a lower level is shown below.



O indicates a generic item and ■ indicates a specific item

The commercial family (entity fami) considered here is BALL POINT PEN

The commercial modules (i.e. entity modu) are holder and top. While ballpoint pen is a generic item indirectly (because it is formed of generic items at lower levels), holder and top are direct generic items.

Generic bill of material (An example)

Parent co-item	Child co-item	Quantity	Unit
Ball point	holder	1	no
Ball point	top	1	no

The parameters and the corresponding parameter values are shown below. There are two parameters, holder colour (denoted as P_1) and top shape (P_2)

Parameter	Holder colour (P_1)	Top shape (P_2)
Parameter Values	Red (PV_1) Black (PV_2) Green (PV_3)	Round (PV_4) Square (PV_5)

The various specific items are as follows

Red holder (S_2), Black holder (S_2) and Green holder (S_3) are for the generic item holder and Round top (S_4) and square top (S_5) for the generic item top.

Conditions

Parameter value	Specific item	Truth value	Status
PV_1	S_1	T	A
PV_2	S_2	T	A
PV_3	S_3	T	A
PV_4	S_4	T	A
PV_5	S_5	T	A

Constraints

O-parameter value	S-parameter value	Truth value
PV_1	PV_2	F
PV_1	PV_3	F
PV_1	PV_3	F
PV_4	PV_5	F

<u>Selection</u>

Let the selection for the parameters be : red (PV 1 is true) for HOLDER COLOR and round (PV_4 is true) for TOP SHAPE. S_1 and S_4 are the selected specific items.

<u>Constraints</u>

Once red is chosen for the parameter HOLDER COLOUR (ie PV_1 is chosen), PV_2 and PV_3 are unqualified for choice. Similarly PV_5 (i.e. parameter value for square) is not qualified for choice once parameter value PV_4 (i.e. for round) is chosen for the parameter TOP SHAPE.

APPENDIX C

AN EXAMPLE TO HIGHLIGHT THE MRP CALCULATION

A simple example is shown to highlight the various MRP process.

Item 1

Period	1	2	3	4
Gross requirements				100
Scheduled receipts				
Projected on-hand	20	20	20	0
Net requirements				80
Planned order release			80	

Netting
100 - (20 + 0)
= 80

Offsetting
by lead time
(for item 1, lead time is 1)

Exploding
80 x 2 = 160 (no of item 2 required)
= 160 (for 1 unit of item 1)

Item 1

Period	1	2	3	4
Gross requirements			160	
Scheduled receipts				
Projected on-hand	10	10	0	
Net requirements			150	100
Planned order release		150		

Netting
160 - (10 + 0)
= 150

Offsetting

APPENDIX D

AN EXAMPLE TO HIGHLIGHT THE CRP CALCULATION

Suppose item A takes 0 20 hrs/unit at capacity Center X, one period after. For period 2 planned order quantity of 40, the capacity center X is utilized for 8 hrs (40x0.2 hrs) on period 3. Similar calculation is done for other periods and for other capacity centers. Again, the similar calculation is done for other items also.

CRP for center X for item

1	2	3	4	5	6
40	40	40		40	40

As far as database is concerned, whatever database that are existing in the MRP system is sufficient for CAP calculation.However for what if analysis, additional database are required for storing alternate plans.

APPENDIX-E

LIST OF ENTITIES AND ITS ATTRIBUTES

The various entities, their attributes and their owner function are given in the following list. The database table name is shown on the left most part. For instance, the plant_t is the database table name of the entity plant. On its right, the explanantion of the entity as well as its owner function are shown. After this, there is a line. Below the line, the various attributes, with its name, explanation and its field type are listed. A star indicates that the particular attribute is a key to the entity.

plant_t	plant or product division	general mgmt	
1. plant#	plant id	varchar(7)	*
2. name	plant name	varchar(15)	
3. location	plant location	varchar(20)	
4. address	plant address	varchar(30)	
5. state	state	varchar(15)	
6. nation	nation	varchar(15)	
7. tele#	telephone no.	varchar(15)	
8. docu#	documentation	varchar(7)	
9. decsr	description	varchar(60)	

g_cap_con_t	global capacity constraint	plant mgmt.	
1. plant#	plant id no	varchar(7)	*
2. g_cap#	global capacity id	varchar(7)	*
3. cap_name	name of cap_constraint	varchar(15)	
4. max_cap	maximum capacity	float	
5. unit_of_mea	unit of measure	varchar(15)	

g_cap_req_t	global capacity requirement	process planning	
1. plant#	plant id no	varchar(7)	*
2. g_cap#	capacity id	varchar(7)	*
3. item#	item id	varchar(7)	*

g_cap_plan_t	global capacity plan	MPP	

1. plant#	plant id	varchar(7)	*
2. g_cap	global capa. con. id	varchar(7)	*
3. lt_per#	long term period id	varchar(7)	*
4. quant	quantity of capa. req.	float	

lt_per_t	Long term period	MPP	
1. lt_per	long term period	varchar(7)	*
2. startdate	starting date	date	
3. enddate	ending date	date	
4. workday	no. of working days	integer	
5. ill_rate	expected absenteism	float	

mpp_t	master production plan	MPP	
1. lt_per	long term period id	varchar(7)	*
2. item#	item id	varchar(7)	*
3. serial#	serial no of the plan	integer	*
4. plan_quant	planned quantity	float	
5. cur_flag	whether currently used	char(1)	

f_ord_t	MPS item firmed up order	MPP	
1. order#	order no.	varchar(7)	*
2. item#	item id	varchar(7)	
3. lt_per#	long term period id	varchar(7)	
4. quant	quantity ordered	float	
5. firm_date	firmed date	date	
6. deli_date	order completion date	date	

r_ord_t	released order	PMC	
1. order#	order id	varchar(7)	*
2. rele_date	released date	date	
3. id#	released person id	varchar(7)	

c_ord_t	completed order	PMC	
1. order#	order no	varcahr(7)	*
2. quant	finished quant	float	
3. fini_date	date of order completion	date	
4. c_flag	whether closed	char(1)	

hedg_data_t	hedging information	PMC	
1. lt_per#	long term period id	varchar(7)	*
2. item #	item id	varchar(7)	*
3. serial#	serial no.	integer	*
4. h_lt_per#	l_t_period id for hedging	varchar(7)	*
5. hedge_quant	hedged quantity	flaot	

prod_targ_t	production target	general mgmt.	
1. plan#	planning module id	varchar(7)	*
2. lt_per#	long term period id	varchar(7)	*
3. plant#	plant no.	varchar(7)	*
4. max_t_rate	max:target rate	float	
5. opt_t_rate	opt:target rate	float	
6. min_t_rate	min:target rate	float	

item_fore	item forecast	forecast	
1. item#	item no.	varchar(7)	*
2. lt_per#	long term period id	varchar(7)	*
3. quant	quantity	flaot	
4. pro_quant	management pro. quant.	float	
5. docu#	documention id	varchar(7)	

prod_unit	production unit (PU)	plant mgmt.	
1. prun#	PU id	varchar(7)	*
2. name	PU name	varchar(15)	
3. descr	PU short description	varchar(60)	
4. docu#	documentation id	varchar(7)	
5. no_emp	no of employees	integer	

c_cap_con	critical capacity constraint	plant mgmt.	
1. prun#	PU id	varchar(7)	*
2. cap#	capacity id.	varchar(7)	*
3. name	name of capacity con.	varchar(15)	
4. max_cap	maximum capacity	float	
5. unit_of_mea	unit of measure	varchar(15)	

c_cap_req	critical capacity requirement	process planning	
1. prun#	PU id	varcahr(7)	*
2. cap*	capacity id .	varchar(7)	*
3. item#	item id	varchar(7)	*
4. quant	quantity/unit of item	float	

c_cap_plan	critical capacity plan	PMC	
1. prun#	PU id	varchar(7)	*
2. cap#	cap. id	varchar(7)	*
3. mt_per#	medium term period id	varchar(7)	*
4. a_quanty	available quantity	float	
5. load	load due mt plan	float	

c_sh_op	closed shop operation	SFC	

1.	shop_order#	shop operation id	varchar(7)	*
2.	item_order#	item order id	varchar(7)	*
3.	act_set	actual set time	float	
4.	act_wait	actual wait time	float	
5.	act_queue	actual queue time	float	
6.	act_move	actual move time	float	
7.	scrap	actual scrap	float	

	batch_info	batch information	SFC	
1.	batch#	batch id	varchar(7)	*
2.	shop_orde#	shop order id	varchar(7)	
3.	item_orde#	item order id	varchar(7)	
4.	quant	quantity	float	
5.	quality	quality remark	varchar(30)	
6.	pick#	picking ticket id	varchar(7)	
7.	rece_date	date of receipt	date	
8.	insp_date	date of inspection	date	
9.	rej_quant	quantity rejected	float	

	mat_iss	material issue	SFC	
1.	shop_oper#	shop operation id	varchar(7)	*
2.	item_orde#	item irder id	varchar(7)	*
3.	item#	item id	varchar(7)	*
4.	quant	quantity	float	
5.	pick#	picking ticket id.	varchar(7)	

	used_mat	used material	SFC	
1.	shop_oper#	shop operation id	varchar(7)	*
2.	item_orde#	item order id	varchar(7)	*
3.	item#	item id	varchar(7)	*
4.	quant	quantity used	float	
5.	pick#	return picking ticket no	varchar(7)	

	empl_assi	employee assignment	SFC	
1.	shop_oper#	shop operation id	varchar(7)	*
2.	item_orde#	item order id	varchar(7)	*
3.	id#	employee id	varchar(7)	*
4.	shift	shift no	integer	
5.	docu#	documentation id	varchar(7)	

	sub_ord	subcontract order	SFC	
1.	sub_orde#	subcontract id	varchar(7)	*
2.	item_orde#	item order id	varchar(7)	*
3.	vend#	vendor id	varchar(7)	
4.	iss_date	issue date	date	
5.	due_date	due date	date	

6. late_date	max. lateness allowed	date	
7. penalty	penalty for lateness	money	
8. unit_o_pen	unit of penalty	varchar(7)	
9. iss_id	issued person id	varchar(7)	

r_item_ord_t	released item order	SFC	
1. item_orde#	item order no.	varchar(7)	*
2. rele_date	released date	date	
3. procplan#	process plan id	varchar(7)	
4. sch_date	scheduled date	date	

c_item_ord_t	closed item order	SFC	
1. item_orde#	item order no.	varchar(7)	*
2. clo_date	closed date	date	
3. quant	quantity finished	float	

shop_oper_t	shop operation	SFC	
1. shop_oper#	shop operation id	varchar(7)	*
2. item_orde#	item order id	varchar(7)	*
3. oper#	operation no.	varchar(7)	
4. created	date of creation	date	
5. sch_date	scheduled date	date	
6. l_sch_date	latest schedule date	date	
7. allo_scrap	allowable scrap	float	
8. equip#	equipment id	varchar(7)	
9. tool#	tool id	varchar(7)	
10. tling#	tooling id	varchar(7)	
11. nc_prog#	nc_prog	varchar(7)	
12. descr	description	varchar(7)	

item_orde_t	item_orde	PMC	
1. item_orde#	item order id	varchar(7)	*
2. item#	item id	varchar(7)	
3. quant	quantity	float	
4. startdate	starting date	date	
5. enddate	ending date	date	
6. pegg#	pegging id	varchar(7)	

m_it_or_t	MPS item order	PMC	
1. item_orde#	item order no	varchar(7)	*
2. orde#	MPS firm order no	varchar(7)	*

f_it_or_t	firmed item order	PMC	
1. item_orde#	item order no.	varchar(7)	*
2. firm date	firmed date	date	

3. quant	quantity	float	
4. sch_date	scheduled receipt date	date	

mt_per_t	medium term period	SFC	
1. mt_per#	medium term period id.	varchar(7)	*
2. startdate	starting date	date	
3. enddate	ending date	date	
4. workday	no. of workind days	integer	
5. ill_raye	expected absenteism	float	

pegg_info_t	pegging information	PMC	
1. pegg#	pegging id	varchar(7)	*
2. item#	item id.	varchar(7)	*
3. source#	source for pegging	varchar(7)	
4. kind	kind of source	char(1)	
5. quant_req	required quantity	float	
6. quant_alloc	allocated quantity	float	
7. alloc_date	allocated date	date	

customer_t	customer details	order processing	
1. cust#	customer id	varchar(7)	*
2. name	name of the customer	varchar(15)	
3. cust_addr	address	varchar(60)	
4. city	city	varchar(15)	
5. state	state	varchar(15)	
6. nation	nation	varchar(15)	
7. tele#	telephone	varchar(15)	
8. ship_addr	address for shipping	varchar(40)	
9. cont_addr	contact person	varchar(15)	
10. cur_bal	current balance	money	
11. tax_rate	tax rate	float	
12. bal_forw	balance forward	money	
13. dtladeb	date of last debit	date	
14. amladeb	amt.of last debit	money	
15. lainvo#	last invoice id	varchar(7)	
16. ytd_sale	year to day sales	float	
17. ytd_pay	year to day payment amt.	money	
18. cre_lim	credit limit	money	
19. descr	description	varchar(60)	

invoice_t	invoice preparation	order processing	
1. invo#	invoice id	varchar(7)	*
2. cust#	customer id	varchar(7)	
3. date	date of issue	date	
4. docu#	documentation	varchar(7)	
5. ship_addr	address for shipping	varchar(40)	
6. c_flag	closed or not	char(1)	
7. descr	description	varchar(60)	

receive_t	receving information	order processing	
1. cust#	customer id	varchar(7)	*
2. ship#	shipment id	varchar(7)	*
3. acc_quant	accepted quantity	float	
4. ret_quant	return quantity	float	
5. date_rece	date receipt	date	
6. date_info	date of informed	date	
7. letter_no	letter no from customer	varchar(20)	
8. docu#	documentation id	varchar(7)	

shipment_t	shipment	order processing	
1. ship#	ship#	varchar(7)	*
2. orno#	customer item order id	varchar(7)	
3. pick#	picking ticket id	varchar(7)	
4. quant	quantity	float	
5. date	date of shipment	date	
6. bill#	bill id	varchar(7)	
7. docu#	documentation	varchar(7)	

cu_it_ord	customer item order	order processing	
1. orno#	customer item order no.	varchar(7)	*
2. invo#	invoice id	varchar(7)	
3. item#	item no.	varchar(7)	
4. quant	quantity	float	
5. unit_price	unit price	money	
6. ship_addr	ship_to address	varchar(40)	
7. c_flag	whethered closed	char(1)	

c_pu_ord_t	closed purchase order	purchase	
1. purc_orde#	purchase order id	varchar(7)	*
2. vend#	vendor id	varchar(7)	
3. item#	item id	varchar(7)	
4. quant	quantity	float	
5. deli_date	delivary date	date	
6. late_quant	late quantity	float	
7. scrap	scrap quantity	float	
8. rework	reworked quantity	float	
9. rewo_cost	rework cost	money	
10. rece_quant	recived quantity	float	
11. price	price	money	
12. penalty	penalty foe lateness	money	
13. discount	discount	float	
14. tax_rate	tax rate	float	
15. docu#	documentation	varchar(7)	

purc_orde_t	purchase release order	purchase

1. vend#	vendor id	varchar(7)	
2. purc_orde#	purchase release order id	varchar(7)	*
3. stoc_room#	delivery location	varchar(7)	
4. iss_date	issue date	date	
5. deli_date	delivary date	date	
6. due_date	date to be supplied	date	
7. sign_id#	person signed id#	varchar(7)	
8. price	price	money	
9. discount	discount	float	
10. taxrate	tax rate	float	
11. bankdetl	bank details	varchar(30)	
12. penalty	penalty for lateness	varchar(7)	
13. docu#	documentation	varchar(7)	

pu_it_rel_t	purchase order-item order rel.	purchase	
1. purc_orde#	purchase order id	varchar(7)	*
2. item_orde#	item order id	varchar(7)	*
3. quant	quantity	float	

vendor_t	vendor details	purchase	
1. vend#	vendor id	varchar(7)	*
2. name	name of vendor	varchar(15)	
3. descr	description	varchar(60)	
4. v_addr	vendor address	varchar(30)	
5. tele#	telephone	varchar(15)	
6. contact	contact person	varchar(15)	
7. cur_bal	current balance	flaot	
8. dtlacre	date of last credit	date	
9. amtlacre	amt. of last credit	money	
10. dtladeb	date of last debit	date	
11. amtladeb	amt.of last debit	money	
12. la_invo#	last_invoice	varchar(7)	
13. ytd_amt.	year to day amount	money	
14. ytd_pay	year to day payment	money	
15. eval_code	evaluation code	varchar(5)	

supp_item_t	supplied item	purchase	
1. item#	item id	varchar(7)	*
2. vend#	vendor id	varchar(7)	*
3. plant#	plant id	varchar(7)	*
4. avg_sp	avg_selling price	money	
5. last_sp	last_selling_price	money	
6. leadtime	leadtime	float	
7. last_lt	last_leadtime	float	
8. la_puor#	last order id	varchar(7)	

procure_t	procurement	purchase	
1. purc_orde#	purchase order id	varchar(7)	*
2. rece_date	received date	varchar(7)	*

3. quant	quantity	float	
4. bal_due	balance due	float	
5. amt_due	amount due	money	
6. inspect#	inspection	varchar(7)	
7. scrap	scrap	float	
8. ret_quant	returned quantity	float	

quotation _t	quotation information	purchase	
1. quto#	quotation id	varchar(7)	*
2. quant	quantity	float	
3. iss_date	issue date	date	
4. due_date	date of submission	date	
5. open_date	date of opening	date	
6. docu#	documentation	varchar(7)	

quto_vend_t	quotation-vendor relationship	purchase	
1. quto#	quotation id	varchar(7)	*
2. vend#	vendor id	varchar(7)	*
3. disp_date	dispatch_date	date	
4. recei_date	received_date	date	

itor_quto_t	quotation item order relationship	purchase	
1. quto#	quotation id	varchar(7)	*
2. item_orde#	item order id	varchar(7)	*
3. quant	quantity	float	
4. sub_flag	sub contract flag	char(1)	

invent_t	inventory status and location	invent mgmt.	
1. item#	item id	varchar(7)	*
2. stoc_room#	stock room	varchar(7)	*
3. bin#	bin#	varchar(7)	*
4. abc_code	abc code	char(1)	
5. stor_capa	max. storage capacity	float	
6. quant	quantity	float	
7. mtd_rece	month today quantity	float	
8. ytd_rece	year today quantity	float	
9. l_item_orde#	last item order id	varchar(7)	
10. inspect#	inspection	varchar(7)	

transact_t	inventory transaction	invent mgmt.	
1. pick#	picking ticket id	varchar(7)	*
2. date	date	date	
3. sour_ord#	source order id	varchar(7)	
4. quant	quantity	float	
5. iss_id#	issued person id	varchar(7)	
6. rece_id#	received person id	varchar(7)	
7. sour_loca#	starting location id	varchar(7)	

8. dest_loca#	ending location id	varchar(7)	

prod_unit_t	production unit	plant mgmt.	
1. prun#	prod. unit id	varchar(7)	*
2. name	name of prod. unit	varchar(7)	
3. descr	short description	varchar(7)	
4. docu#	documentation	varchar(7)	
5. no_emp	no of employees	integer	

capa_cent_t	capacity centre	prod. unit mgmt.	
1. cace#	capacity center id	varchar(7)	*
2. name	name of capa. cent	varchar(15)	
3. type	type code	varchar(15)	
4. foreman#	id# of foreman	varchar(7)	
5. docu#	documentation	varchar(7)	
6. descr	short description	varchar(60)	

equipment_t	equipment details	prod_unit mgmt.	
1. equip#	equipment id	varchar(7)	*
2. name	name of equip.	varchar(15)	
3. descr	short description	varchar(60)	
4. cace#	capacity centre	varchar(7)	
5. location	location details	varchar(25)	
6. price	price	money	
7. lmaint#	last maint. id	varchar(15)	
8. nohrs	no of hrs worked	float	
9. cuhrs	cumulative hrs worked	float	
10. docu#	documentation	varchar(7)	
11. supp#	supplier id	varchar(7)	
12. equip_type	equipment type code	varchar(10)	
13. max_load	max. load	float	
14. max.pow	max. power	float	
15. max.tor	max. torque	float	
16. opt_load	opt. load	float	
17. tableng	table length	float	

tool_t	tool details	prod_unit mgmt.	
1. tool#	tool id	varchar(7)	*
2. name	name of the tool	varchar(15)	
3. descr	short description	varchar(60)	
4. cace#	capacity centre mostly used	varchar(7)	
5. mat_code	material code	varchar(10)	
6. price	price	money	
7. docu#	documentation	varchar(7)	
8. prun#	prodction unit	varchar(7)	

tooling_t	tooling details	prod_unit mgmt.	
1. tling#	tooling id	varchar(7)	*

2. name	name of the tooling	varchar(15)	
3. descr	short description	varchar(60)	
4. cace#	capacity centre mostly used	varchar(7)	
5. mat_code	material code	varchar(10)	
6. price	price of tooling	money	
7. docu#	documentation	varchar(60)	
8. prun#	prodction unit	varchar(7)	

mat_ha_eq_t	material handling equipment	prod_unit mgmt.	
1. equip#	mat. handling equip. id	varchar(7)	*
2. name	mat. handling equip. name	varchar(15)	
3. descr	short description	varchar(60)	
4. prun#	prodction unit	varchar(7)	
5. maxload	maxmiun load	float	
6. optload	optmium load	float	
7. mequip_type	mat.handling equip. type	varchar(10)	
8. lmaint#	last maintenance id	varchar(7)	
9. price	price	money	
10. supp#	supplier id	varchar(7)	
11. nohrs	no of hrs	float	
12. cuhrs	cumulative hrs	float	

break_hist_t	breakdown history	maintenance	
1. break#	breakdown id.	varchar(7)	*
2. equip#	equipment id	varchar(7)	*
3. occ_date	occurance date	date	
4. rep_findate	repair finished date	date	
5. docu#	documentation	varchar(7)	

pre_maint_t	preventive maintenance details	maintenance	
1. equip#	equipment id	varchar(7)	*
2. pmaint#	pre. maintenance id	varchar(7)	*
3. name	name of the maint.operation	varchar(15)	
4. descr	short description	varchar(60)	
5. time	duration	float	
6. when	when to do	float	
7. docu#	documentation	varchar(7)	

pre_ma_it_t	preventive maintenance item req.	maintenance	
1. pmaint#	pre. maintenance id	varchar(7)	*
2. item#	item id.	varchar(7)	*
3. quant	quantity	float	

act_main_t	actual maintenance	maintenance	
1. equip#	equipment id	varchar(7)	*
2. maint#	maintenance action id	varchar(7)	*
3. name	name of maintenance action	varchar(10)	
4. descr	short description	varchar(60)	

5. startdate	starting date	date	
6. enddate	ending date	date	
7. foreman#	foreman id	varchar(7)	
8. source#	whether pre. or break.	varchar(7)	
9. docu#	documentation	varchar(7)	

maint_item_t	maintenance items used	maintenance	
1. maint#	maintenance action id	varchar(7)	*
2. item#	item id	varchar(7)	*
3. quant	quantity	float	

maint_empl_t	maintenance employee assigned	maintenance	
1. maint#	maintenance action id	varchar(7)	*
2. id#	employee id	varchar(7)	*
3. startdate	start time	date	
4. enddate	end time	date	
5. eval	evalation code	varchar(5)	

proc_plan_t	process plan information	process planning	
1. item#	item id.	varchar(7)	*
2. procplan#	process plan id.	varchar(7)	*
3. min batch	min. batch size	float	
4. max batch	max. batch size	float	
5. opt batch	opt. batch size	float	
6. name	name of process plan	varchar(15)	
7. docu#	documentation	varchar(7)	
8. plan_id#	planner code	varchar(7)	
9. rel_date	release date	date	
10. stdtime	std production/unit	float	
11. stdtime	std labour/unit	float	
12. stdcost	std cost/unit	money	
13. revdate	revision date	date	
14. descr	short description	varchar(60)	

norm_oper_t	normative operation	process planning	
1. oper#	operation id.	varchar(7)	*
2. seq_no	sequence no.	integer	*
3. procplan#	process plan id.	varchar(7)	*
4. name	name of operation	varchar(7)	
5. cace#	capacity centre id.	varchar(7)	
6. pre_oper#	previous operation	varchar(7)	
7. next_oper#	next operation	varchar(7)	
8. s_set_time	std. set time	float	
9. s_que_time	std. queue time	float	
10. s_run_time	std. run time	float	
11. descr	description	varchar(7)	

material_t	material details	process planning

1. mat_code	material code	varchar(10)	*
2. name	name of the material	varchar(15)	
3. descr	description	varchar(60)	
4. minhard	minmium hardness	float	
5. maxhard	maxmium hardness	float	
6. unit_of_meas	unit of measure	varchar(10)	
7. pretreat	pretreatment	varchar(40)	
8. yieldstr	yield strength	float	
9. tensistr	tensile strength	float	
10. elong	elongation	float	
11. cutstiff	cut stiffness	float	
12. microstr	microstructure	varchar(60)	

eq_tl_it_t	equip. tooling item relationship	process planning	
1. equip#	equipment id.	varchar(7)	*
2. tling#	tooling id.	varchar(7)	*
3. item#	item id.	varchar(7)	*
4. max_len	maxmium length	float	
5. max_bre	maxmium breadth	float	
6. max_hei	maxmium height	float	
7. max_load	maxmium load	float	

eq_to_ma_t	equipment tool material rel.	process planning	
1. equip#	equipment id.	varchar(7)	*
2. tool#	tool id.	varchar(7)	*
3. mat_code	material code	varchar(10)	*
4. cuttype	cut type	varchar(15)	
5. maxfeed	max feed rate	float	
6. optfeed	opt feed rate	float	
7. maxdep	max. depth of cut	float	
8. optdep	opt. depth of cut	float	
9. maxcutspe	max. speed of cut	float	
10. optcutspe	opt. speed of cut	float	
11. maxtolife	max. tool life	float	
12. avgtolife	avg. tool life	float	
13. coolant	coolant	varchar(20)	
14. tool_hold#	tool_hloder	varchar(7)	

it_meq_t	item mat. handling equip.rel.	process planning	
1. item#	item id.	varchar(7)	*
2. equip#	equipment id.	varchar(7)	*
3. t_quant	transfer quantity	float	
4. s_lo_time	std. load time	float	
5. s_unlo_time	std. unload time	float	
6. maxwt	max. weight	float	

noop_faci_t	norm. oper.and facility rel.	process planning	
1. oper#	operation id.	varchar(7)	*

2. tool#	tool id.	varchar(7)	*
3. tling#	tooling id.	varchar(7)	*
4. equip#	equipment id.	varchar(7)	*
5. stdtime	std. time for set up	float	
6. nc_prog#	nc_prog details	varchar(7)	
7. inspect#	inspection details	varchar(7)	

consume_t	req. mat. for yhe operation	process planning	
1. oper#	operation id.	varchar(7)	*
2. p_item#	parent item id.	varchar(7)	*
3. c_item#	child item id.	varchar(7)	*
4. quant	quantity	float	
5. scrap	allowable scrap	float	

fami_t	commercial family	product definition	
1. co_item#	commercial item	varchar(7)	*
2. ag_demand	aggregated demand	integer	

paramete_t	parameter for comm. family	product definition	
1. para#	parameter id.	varchar(7)	*
2. co_item#	comm. item id.	varchar(7)	
3. name	name of the parameter	varchar(15)	
4. descr	short description	varchar(60)	
5. created	date of creation	date	
6. modified	date of modification	date	

par_value_t	parameter value	product definition	
1. pave#	parameter value id.	varchar(7)	*
2. para#	parameter id.	varchar(7)	
3. name	name of parameter value	varchar(15)	
4. descr	short description	varchar(60)	
5. mult_f	multification factor	integer	

constra_t	constraint	product definition	
1. o_pave#	object parameter value id	varchar(7)	*
2. s_pave#	subject parameter value id	varchar(7)	*
3. true_val	truth value	varchar(5)	
4. created	date of creation	date	
5. modified	date of modification	date	
6. descr	description	varchar(60)	

condit_t	condition	product definition	
1. pave#	parameter value id.	varchar(7)	*
2. item#	item id.	varchar(7)	*
3. true_val	truth value	varchar(5)	

4. status	and/or statųs of condiion	varchar(3)
5. created	date of creation	date
6. modified	date of modification	date
7. descr	short description	varchar(60)

sp_item_t	specific item	product definition	
1. item#	item id.	varchar(7)	*
2. co_item#	comm.item id.	varchar(7)	
3. quant	quantity	integer	
4. unit	unit of measure	varchar(10)	
5. comm_name	comm.name of sp.item.	varchar(15)	
6. descr	short description	varchar(60)	

co_item_t	commercial item	product definition	
1. co_item#	comm.item id.	varchar(7)	*
2. kind	kind of comm.item	varchar(6)	
3. name	name of comm. item	varchar(15)	
4. docu#	documentation	varchar(7)	
5. descr	short description	varchar(60)	
6. created	date of creation	date	
7. modified	date of creation	date	

gene_bom_t	generic bill of material	product definition	
1. p_co_item#	parent comm. item id.	varchar(7)	*
2. c_co_item#	child comm. item id.	varchar(7)	*
3. quant	quantity	float	
4. unit	unit of measure	varchar(10)	
5. effstrdate	start date	date	
6. effenddate	end date	date	
7. created	date of creation	date	
8. modified	date of modified	date	
9. descr	short description	varchar(60)	

modu_t	comm. module	product definition	
1. co_item#	comm. item id.	varchar(7)	*
2. ex_price	external price	money	
3. in_price	internal price	money	

stoc_item_t	stock item	product definition	
1. item#	item id.	varchar(7)	*
2. safety_stoc	safety level req.	float	
3. safetyleadtime	safety lead time	float	
4. stoc_kind	kind of stock item	varchar(7)	

cmfp_make_t	comm.item-final prod. rel.	product definition

1. co_item#	comm.item id.	varchar(7)	*
2. fipr#	final prod.id.	varchar(7)	*
3. mix_percent	percent of final prod.	float	
4. created	date of creation	date	
5. modified	date of modified	date	
6. docu#	documentation	varchar(7)	
7. descr	description	varchar(60)	
8. createdid#	created person id.	varchar(7)	
9. modifiedid#	modified person id.	varchar(7)	

cusp_make_t	customer sp. item makeup	product definition	
1. cusp#	cust. sp. item id.	varchar(7)	*
2. item#	item id.	varchar(7)	*
3. quant	quantity	float	
4. descr	short description	varchar(60)	
5. delidate	date of delivary	date	

cusp_prod_t	cust. sp. prod. item	product definition	
1. cusp#	cust. sp. item id.	varchar(7)	*
2. orno#	customer item id	varchar(7)	
3. suppdate	suppy date	date	
4. draw#	drawing id.	varchar(7)	
5. docu#	documentation	varchar(60)	

stan_prod_t	standard final product	product definition	
1. fipr#	final prod. id.	varchar(7)	*
2. brand_name	brand name of the product	varchar(15)	
3. ag_demand	aggregated demand	float	

fipr_prod_t	final product	product definition	
1. fipr#	final prod. id.	varchar(7)	*
2. comm_name	comm.name of prod.	varchar(15)	
3. ex_price	external price	money	
4. in_price	internal price	money	
5. created	date of creation	date	
6. modified	date of modified	date	
7. co_item#	comm. item id.	varchar(7)	
8. descr	description	varchar(60)	
9. standard	stan. or cust. sp. prod.	varchar(3)	

plan_modu_t	palnning module	product definition	
1. plan#	plan. module id.	varchar(7)	*
2. plan_kind	kind of plan. module	varchar(8)	
3. name	name of plan. module	varchar(15)	
4. descr	short description	varchar(60)	
5. created	date of creation	date	
6. modified	date of modified	date	

7. docu#	documentation	varchar(7)	

capl_modu_t	capacity plan.module	product definition	
1. plan#	plan. module id.	varchar(7)	*
2. unit_cost	unit cost	float	

mapl_modu_t	material planning module	product definition	
1. plan#	plan. module id.	varchar(7)	*
2. max_unitprice	max unit price	float	
3. avg_unitprice	avg unit price	float	
4. min_unitprice	min unit price	float	

cfpm_make_t	comm. fami.- plan. modu. rel	product definition	
1. co_item#	comm. item id.	varchar(7)	*
2. plan#	plan. module id.	varchar(7)	*
3. mix_percent	mix percentage	float	
4. created	date of creation	date	
5. modified	date of modified	date	
6. docu#	documentation	varchar(7)	
7. descr	description	varchar(60)	
8. created id#	created person id.	date	
9. modified	modified person id	date	

pmsp_make_t	plan.modu.-stan.final prod.rel	product definition	
1. plan#	plan. module id.	varchar(7)	*
2. fipr#	final prod.id.	varchar(7)	*
3. mix_percent	mix percentage	float	
4. created	date of creation	date	
5. modified	date of modified	date	
6. docu#	documentation	varchar(7)	
7. descr	description	varchar(60)	
8. created id#	created person id.	date	
9. modified	modified person id	date	

mapm_make_t	mat.plan.modu.-purc. item rel.	product definition	
1. plan#	plan.module id.	varchar(7)	*
2. item#	item id.	varchar(7)	*
3. mix_percent	mix percentage	float	
4. mix_descr	mix description	varchar(60)	

mps_item_t	mps item details	product definition	
1. item#	item id.	varchar(7)	*
2. plan#	plan. module id	varchar(7)	
3. mix_percent	mix percentage	float	
4. created	date of creation	date	

5. h_t_fence	hedge time fence	integer	
6. d_t_fence	demand time fence	integer	
5. p_t_fence	plan time fence	integer	

phantom_t	phantom item	product definition	
1. item#	item id.	varchar(7)	*
2. cace#	capacity centre id.	varchar(7)	

product_t	manufactured product	product definition	
1. item#	item id.	varchar(7)	*
2. master	whether mps_item or not	varchar(1)	

stoc_item_t	stock item	product definition	
1. item#	item id.	varchar(7)	*
2. safety_stoc	safety quantity	float	
3. safetyleadtime	saefty lead time	float	
4. stoc_kind	kind of stock item.	varchar(7)	

manu_item_t	manufactured	product definition	
1. item#	item id.	varchar(7)	*
2. prun#	prod_unit id.	varchar(7)	
3. manu_kind	kind of manu_item	varchar(7)	
4. setuptime	set up time	float	
5. runtime	run time	float	
6. movetime	move time	float	
7. queuetime	queue time	float	

purchase_t	purchase item	product definition	
1. item#	item id.	varchar(7)	*
2. lotsize	lot size	float	
3. low_price	lowest price	money	
4. high_price	highest price	money	

item_bom_t	item bill of material	product definition	
1. p_item#	parent item id.	varchar(7)	*
2. c_item#	child item id.	varchar(7)	*
3. quant	quantity	float	
4. unit	unit of measure	varchar(10)	
5. efstddate	start date	date	
6. efenddate	end date	date	
7. descr	description	varchar(60)	
8. assdraw#	drawing details	varchar(7)	
9. created	date of creation	date	
10. modified	date of modified	date	
11. allo_scrap	allowable scrap	flaot	

item_t	item details	product definition	
1. item#	item id.	varchar(7)	*
2. name	name of item	varchar(15)	
3. descr	description	varchar(60)	
4. item_kind	kind of item	varchar(12)	
5. std_cost	standard cost	money	
6. low_lev_code	low level code	integer	
7. draw#	drawing id.	varchar(7)	
8. rele_date	release date	date	
9. planner_code	planner code	varchar(7)	
10. ord_pol_code	order policy code	varchar(7)	
11. docu#	documentation	varchar(7)	
12. lotsize	lot size	float	
13. unit_of_meas	unit of measure	varchar(10)	

counter_t	counter details of various entities	database adm.	
1. counter#	counter id.	varchar(7)	*
2. code	code characater	char(2)	
3. last#	last id	integer	
4. descr	description	varchar(15)	

APPENDIX-F

LIST OF FORMS, FRAMES AND PROGRAMMES

no:	form	frame	programme	explanation (frame for)
			Main frame	
1.	mpcso	mpcsr1	mpcsr1.osq	main frame
2.	mastso	mastsr1	mastsr1.osq	master planning
3.	macoso	macosr1	macosr1.osq	plant material coordination
4.	pucoso	pucosr1	pucosr1.osq	production unit control
			Master planning frame	
5.	plantto	planttr1	planttr1.osq	db table for plant info.
6.	prtato	prtatr1	prtatr1.osq	db table for prod. traget
7.	itfoto	itfotr1	itoftr1.osq	db table for forecast details
8.	orprso	orprsr1	orprsr1.osq	order processing
9.	mppsro	mppsr1	mppsr1.osq	master production planning
10.	bofmso	bofmsr1	bofmsr1.osq	bill of material(BOM)
11.	purchso	purchsr1	purchsr1.osq	purchase
			Order processing	
12.	custto	custtr1	custtr1.osq	db table for customer inform.
13.	invoto	invotr1	invotr1.osq	db table for invoice prepare.
14.	shipto	shiptr1	shiptr1.osq	db table for shipment details
15.	receto	recetr1	recetr1.osq	db table for receive inform.
16.	cuitoto	cuitotr1	cuitotr1.osq	db table for cust. item order
			Master production planning	
17.	gcapto	gcaptr1	gcaptr1.osq	db table for glob. capa. plan
18.	ltprto	ltprtr1	ltprtr1.osq	db table for long term period
19.	mppto	mpptr1	mpptr1.osq	db table for master prod.plan
20.	fordto	fordtr1	fordtr1.osq	db table for firm order detl.
21.	hedgeto	hedgetr1	hedgetr1.osq	db table for hedge data
			Purchase	
22.	vendto	vendtr1	vendtr1.osq	db table for vendor details
23.	suitto	suittr1	suittr1.osq	db table for supplied item
24.	prorto	prortr1	prortr1.osq	db table for pur.rel.ord(PRO)
25.	piorto	piortr1	piortr1.osq	db table for PRO & item order
26.	procto	proctr1	proctr1.osq	db table for procurement
27.	quotto	quottr1	quottr1.osq	db table for quotation
28.	itquto	itqutr1	itqutr1.osq	db table for it.ord.-quot.rel.
29.	qverto	qvertr1	qvertr1.osq	db table for quot.-vend. rel.
30.	cproto	cprotr1	cprotr1.osq	db table for closed pur. ord.
			Bill of material (BOM)	
31.	commso	commsr1	commsr1.osq	BOM for comm. plan.& control
32.	logiso	logisr1	logisr1.osq	BOM for logi. plan.& control
33.	prodso	prodsr1	prodsr1.osq	BOM for prod. plan.& control

82.	itemto	itemtr1	itemtr1.osq	db table for item
83.	itemso	itemsr1	itemsr1.osq	creating a item
84.		itemsr2	itemsr2.osq	displaying a item
85.		itemsr3	itemsr3.osq	editing a item
86.	ibomto	ibomtr1	ibomtr1.osq	db table for item bill of mat
87.	ibomso	ibomsr1	ibomsr1.osq	creating a item bill of mat.
88.		ibomsr2	ibomsr2.osq	displaying a item bill of mat.
89.		ibomsr3	ibomsr3.osq	editing a item bill of mat.

Plant material coordination

90.	prunto	pruntr1	pruntr1.osq	db table for production unit
91.	mcopto	mcoptr1	mcoptr1.osq	mat. coordination planning
92.	prplto	prpltr1	prpltr1.osq	process plan inform.
93.	istto	isttr1	isttr1.osq	inventory storage and trans.
94.	gcacto	gcactr1	gcactr1.osq	db table for global capa.con.
95.	ccacto	ccactr1	ccactr1.osq	db table for crit. capa.con.

Material coordination planning

96.	reorto	reortr1	reortr1.osq	db table for released order
97.	coorto	coortr1	coortr1.osq	db table for completed order
98.	peggto	peggtr1	peggtr1.osq	db table for pegging inform.
99.	itorto	itortr1	itortr1.osq	db table for item order
100.	miorto	miortr1	miortr1.osq	db table for MPS item
101.	fiorto	fiortr1	fiortr1.osq	db table for firmed item ord.
102.	mtprto	mtprtr1	mtprtr1.osq	db table for medium term per.
103.	ccplto	ccpltr1	ccpltr1.osq	db table for crit.capa.plan

process plan information

104.	propto	proptr1	proptr1.osq	db table for process plan
105.	noopto	nooptr1	nooptr1.osq	db table for normative proc.
106.	eqtomato	eqtomatr1	eqtomatr1.osq	db table for equip-tool-mat.
107.	eqtlitto	eqtlittr1	eqtlittr1.osq	db table for equip-tling.-mat.
108.	itmeqto	itmeqtr1	itmeqtr1.osq	db table for item-mat.hand.eq
109.	nofato	nofatr1	nofatr1.osq	db table for normative oper.
110.	coneto	conetr1	conetr1.osq	db table for consume
111.	sureto	suretr1	suretr1.osq	db table for subcont. req.
112.	gcarto	gcartr1	gcartr1.osq	db table for global capa.req.
113.	ccarto	ccartr1	ccartr1.osq	db table for crit.capa.req.

Inventory storage and transaction

114.	invento	inventtr1	inventtr1.osq	db table for inve.stat.& loc.
115.	transto	transtr1	transtr1.osq	db table for inve. transact.

Production unit control

116.	faciso	facisr1	facisr1.osq	db table for facility inform.
117.	maintso	maintsr1	maintsr1.osq	db table for maintenance
118.	sfcso	sfcsr1	sfcsr1.osq	db table for shop floor cont.

Facility information

119.	tointo	tointr1	tointr1.osq	db table for tool inform.
120.	tlingto	tlingtr1	tlingtr1.osq	db table for tooling inform.
121.	equipto	equiptr1	equiptr1.osq	db table for equip.inform.
122.	mequipto	mequiptr1	mequiptr1.osq	db table for mat.hand. equip.
123.	caceto	cacetr1	cacetr1.osq	db table for capa.centre

			Maintenance	
124.	maemto	maemtr1	maemtr1.osq	db table for maint. emplo.
125.	maitto	maittr1	maittr1.osq	db table for maint. item
126.	breakto	breaktr1	breaktr1.osq	db table for breakdown info.
127.	acmato	acmatr1	acmatr1.osq	db table for act.maint.action
128.	prmato	prmatr1	prmatr1.osq	db table for pre.maint.action
129.	pritto	prittr1	prittr1.osq	db table for pre.maint.item
			Shop floor control	
130.	riorto	riotr1	riortr1.osq	db table for rele. item order
131.	ciorto	ciortr1	ciortr1.osq	db table for comp. item order
132.	sordto	sordtr1	sordtr1.osq	db table for subcontract ord.
133.	shopto	shoptr1	shoptr1.osq	db table for shop operation
134.	cshopto	cshoptr1	cshoptr1.osq	db table for com.shop oper.
135.	emasto	emastr1	emastr1.osq	db table for employee assign.
136.	maisto	maistr1	maistr1.osq	db table for material issued
137.	usmato	usmatr1	usmatr1.osq	db table for used material
138.	bainto	baintr1	baintr1.osq	db table for batch inform.

IME-1993-M-CHE-DES